Universität Regensburg
Fakultät für Wirtschaftswissenschaften
Lehrstuhl für Wirtschaftsinformatik I - Informationssysteme

Sustainable Compliance in Identity and Access Management

Dissertation

zur Erlangung des Grades eines Doktors der Wirtschaftswissenschaft, eingereicht an der Fakultät für Wirtschaftswissenschaften der Universität Regensburg

vorgelegt von
Sebastian Groll, M.Sc. with Honors

Berichterstatter:
Prof. Dr. Günther Pernul
Prof. Dr. Susanne Leist

Tag der Disputation: 22.05.2025

To my parents Brigitte and Josef,
and my family Julia and Sophia.

Abstract

Compliance regulations regarding information system security are playing an increasingly important role for organizations. Identity and Access Management (IAM), i.e. the structured management of employees and their access rights, is a powerful means for meeting compliance requirements. However, IAM-related entities and data are subject to constant change. For instance, new employees frequently join or leave organizations and applications are introduced or discontinued. Similarly, the regulatory requirements are continuously altered. Thus, compliance measures in IAM must be implemented in a sustainable and maintainable manner. First steps in this direction are taken by proposing approaches to ensure the data quality of IAM attributes and access control policies. High quality IAM data is a key requirement for meeting compliance guidelines and for implementing compliance measures. Subsequently, two central compliance measures are studied: Access Reviews (AR) and Separation of Duty (SoD). In ARs reviewers have to manually decide which access rights should be assigned to which users. An approach is proposed that assesses the quality of these decisions and thus helps to find structural issues in the execution of AR campaigns. For SoD, i.e. a compliance and security measure that ensures no individual has enough access rights to perform a critical process alone, a structured standalone literature review is conducted. Furthermore, a management approach for SoD rules is proposed, which especially focuses on human understandability and maintainability.

Acknowledgement

During my long and enriching journey towards completing this dissertation, I had the privilege of collaborating with many exceptional colleagues. I am especially pleased that many of these companions have become true friends along the way. It is an honor to express my gratitude to them in the following.

My deepest thanks go to Prof. Günther Pernul, who has consistently made his vast experience and extensive knowledge available to me throughout my journey. Despite his numerous commitments and duties, he always takes the time to support his students whenever needed. His valuable advice and constructive criticism have always encouraged me to think outside the box. Furthermore, I would like to thank my second supervisor, Prof. Susanne Leist, for her support and advice.

A significant role, especially at the beginning of my academic career, was played by my very good friend and colleague Dr. Michael Kunz, who provided outstanding support. Together, we navigated not only professional challenges but also personal ups and downs.

The same can be said for Sascha Kern, with whom I had many fruitful discussions about science and practice, often lasting late into the night and occasionally accompanied by some drinks.

Furthermore, I would like to thank my friend Dr. Ludwig Fuchs, who has supported me with his valuable advice in all my work. He created an environment that has allowed me to turn my hobby into a profession.

Then I would like to thank Alex, who started this journey with me, walking alongside on the same path.

I also extend my thanks to Dr. Matthias Hummer, whose support and inspiration were particularly significant at the beginning of my academic journey.

A special thanks goes to Prof. Dr. Michael Netter, whose passion for research and nearly ideological commitment to a clear and ethical approach have deeply inspired me.

I also want to express my gratitude to all my colleagues at the Chair of Information Systems I and at Nexis GmbH, who traveled this journey alongside me. Special mentions go to Manfred, Fabian, Ludwig, Marietheres, Thomas, and, last but not least, Raphael. I especially wish Raphael great success in his new academic endeavors.

Last but not least, I would like to thank my parents and my family, who had to (and still have to) bear the consequences of my ambitions. Thank you for your understanding. I love you.

Contents

List of Tables

List of Figures

Part I

Overview of the Thesis

1 Introduction

The advancing digitalization has continually confronted organizations with new challenges concerning information system (IS) security, the management of employees and the management of their access rights[1]. Organizations are exposed to external threats such as cyber attacks as well as internal threats such as fraud or data theft by their own employees. Incidents like the Wirecard scandal or whistleblowers like Edward Snowden show the extent to which these dangers can affect people and society. This is especially true for organizations providing essential services or critical infrastructure, like health care, the finance sector or the military. In order to reduce these risks to an acceptable level, countries have established regulatory measures and frameworks concerning the management of employees and their access rights for organizations[2]. Additionally, organizations are obliged to meet security standards that are required by their economic environment. For example, customers or business partners may require certifications to ensure a minimum level of security. Due to the growing number of regulations and the increase in data to be managed, complying with all these regulations is becoming increasingly difficult.

In this context, enterprise Identity and Access Management (IAM) plays a central role. It deals with the structured administration and protection of digital identities [23], i.e. the representations of the employees in the digital world, and the management of their access rights. IAM provides several means to comply with IS security standards[3]. One of the key components of IAM in terms of IS security and compliance is authorization management. In this context, the two Access Control Models (ACM) role-based access control (RBAC) [24] and attribute-based access control (ABAC) [15] have been established. In RBAC, instead of assigning access rights directly, roles are added as intermediary between employees and access rights. A role is assigned to multiple employees and contains multiple permissions. By doing so, the number of assignments significantly drops and a layer of information is added: Roles can carry meaning by name or description e.g. a role named "Software Developers" should be assigned to the developers in an organization and contains the access rights software developers need to do their work. This reduces complexity and the risk of employees having incorrect or missing access rights. In ABAC, access rights are assigned and revoked automatically based on policies that consist of rules. These rules evaluate the attributes of employees and access rights, as well as the context (e.g. time or location) and determine if an employee can have a specific access right or not. For example an employee with the job title "Nurse" and work schedule "Night Shift" could have access to the patient records of a hospital at night. Regarding compliance, ABAC and RBAC play an important role in adhering to the Principle of

[1] The term "access right" is used as synonymous with the term "permission" in this dissertation. It represents the authorization given to a user to carry out a single specific transaction on a system.

[2] Common examples are SOX [20], BSI [21], Basel III [1], BAIT [7], HIPAA [26], or VAIT [4].

[3] Note that the set of IAM-related compliance measures discussed within this dissertation is, of course, not complete. Numerous other IAM measures are important for compliance, including, among others, identity life-cycle management, authorization concepts, authentication, and reporting.

Least Privilege [24, 8], which states that employees should not possess more access rights than they actually need for their work.

The success of ACMs heavily relies on data quality [18, 17], especially ABAC relies on attribute accuracy. For instance, if an intern at a hospital mistakenly gets the job title "Nurse," she could gain access to critical patient records. Another challenge is maintenance and sustainability. Roles and ABAC policies become outdated over time, because organizations are subject to constant change. For example, new employees join the company, change their role or leave. Also new applications and access rights are added over time or change their functionality. Furthermore, external circumstances, such as the security situation or changes in legal requirements necessitate the continuous maintenance of ACMs.

A closely related compliance measure in this context is the conduction of Access Reviews (AR). ARs are intended to enforce the Principle of Least Privilege and ensure accountability for the access right assignments. In an AR, a responsible person is required to manually check all access right assignments and decide whether they are correct or not. The responsible persons are usually the supervisors of the respective employees, but may also be the responsibles for the access rights (e.g. the administrator of an application). ARs are carried out in regular campaigns and the results have to be documented for auditing and compliance purposes. While auditors often require ARs, they also have some major drawbacks: they are very time-consuming and the benefit for IS security heavily depends on how carefully they are executed. In order to ensure sustainable security, it is important to verify the accuracy of AR executions. Otherwise, a false sense of security could emerge.

Another compliance requirement that can be enforced using IAM is Separation of Duty (SoD). SoD is a security principle designed to reduce fraud and insider misuse by ensuring that no single employee has full control over all aspects of a critical task or process. For example, if an accountant is permitted to both record and approve expenses, she could submit and approve fake expense reports, funneling money into her own account. IAM can be used to enforce SoD by ensuring that no employee has all access rights to perform those critical tasks. Therefore, SoD rules[4] must be established that restrict ACMs regarding the assignment of access rights. One way to set up a corresponding SoD rule is to define a set of access rights necessary to execute a critical task, along with the minimum number of employees required to collectively possess all these access rights [24]. This ensures that at least the defined number of employees must work together in order to execute the task. A simpler approach is to define mutual exclusive access rights or roles, which an employee must not possess together [19]. Regardless of the type of specification, it is important that the rules cover the respective compliance requirements. Since this requires both legal knowledge and information about access rights and applications, the relevant knowledge is distributed throughout the company. Another challenge, particularly

[4]The terms SoD policy and SoD rule are often used interchangeably. Nevertheless, SoD policy additionally refers to higher-level policies (e.g., in the form of documents and legal texts), while SoD rule only refers to specific SoD exclusions (e.g., two mutual exclusive access rights).

in terms of sustainability, is that compliance requirements, access rights and applications are subject to continuous change, requiring a continuous adaptation of the SoD rules. To summarize, a major challenge when ensuring compliance in IAM and implementing IAM-related compliance measures is to establish maintainability and sustainability. This is especially important due to the fluctuations in IAM data and changing compliance requirements. This dissertation aims to sustainably improve compliance in IAM, with a particular focus on ACMs, ARs, and SoD.

The remainder of this dissertation is structured as follows. Section 2 formulates the primary research question as well as three sub-research questions and corresponding research challenges (RCs) that are addressed within this dissertation. Section 3 outlines the research methodology applied to the publications belonging to this thesis. In Section 4 the results of the respective publications are presented. First, an overview and the connections between the publications are provided. Then the individual publications are presented in detail in the context of this dissertation. The complementary publications that do not directly contribute to this work are listed at the end of Section 4. Section 5 concludes Part I of the thesis, by summarizing and discussing the results and suggesting recommendations for further research. Finally, the original publications are printed in Part II of the dissertation.

2 Research Questions

As previously stated, it is a major challenge in IAM to implement compliance measures that are sustainable and maintainable despite the fluctuating nature of IAM-related data and compliance requirements. This leads to the following primary research question for this dissertation:

RQ: *How to improve IAM related compliance in a sustainable way?*

To answer the research question, this dissertation is divided into three research areas. First, the quality of IAM data, especially the quality of compliance-relevant access control policies (ACP)[5], has to be ensured. This topic is addressed in the first research area. Afterwards, two central compliance measures are focused on: The first focus area addresses ARs, a central compliance measure for maintaining ACPs which is often required by regulations and auditors. The second focus area is on SoD policies, which are ACPs designed to ensure compliance with SoD requirements. For each research area, a research question is derived, containing several research challenges (RCs). The answers to these research questions represent the substance of this dissertation. In the following, the research areas and the corresponding research questions are described in detail.

[5]The term access control policy is very comprehensive and includes all policies and rules that regulate the access to (digital) resources. Typical examples of ACPs are roles in RBAC, ABAC policies or SoD policies.

Foundation: Data and policy quality in IAM

In IAM, ACPs regulate who receives access to which resources. ACPs therefore represent a central mechanism to comply with the Principle of Least Privilege, which is required by most compliance standards[6]. Consequently, it is necessary to ensure the proper maintenance and quality of the ACPs and the data they operate on, which typically comprise attributes of IAM-relevant entities[7]. This leads to the first sub-research question:

RQ 1: *How to ensure attribute data and ACP quality within IAM?*

To answer this question, three research challenges are derived. First, a generic conceptual IAM model has to be developed that is compatible with a broad spectrum of standards and regulations *(RC 1.1)*. This model must establish the relationships between the most important IAM-relevant entities and can thus serve as a basis for further research. The application of such a generic IAM model extends beyond the area of data and policy quality, as it can also serve as a foundation for AR or SoD approaches. Second, the term *quality* in relation to IAM relevant data and ACPs must be clearly defined *(RC 1.2)*. This implies identifying the relevant quality criteria for an effective IAM. And finally, approaches must be found and developed that are capable to maintain and improve ACP and attribute quality *(RC 1.3)*. In order to achieve this in a sustainable way, a structured approach that targets continuous quality management (e.g., through tool support) is needed.

Compliance Focus 1: Access Reviews

ARs are typically requested by auditors and conducted in a regular, campaign-like manner. During an AR, several responsible persons (either those responsible for employees or access privileges) must manually inspect the privilege assignments and mark the wrong / excessive ones. This contributes to security, the Principle of Least Privilege and also creates a responsible person for the privilege assignments. Assuming that all responsible persons made correct decisions, there should be no excessive privilege assignments left after a comprehensive AR campaign. However, there are many reasons why incorrect decisions are made especially in ARs: Often, the high number of decisions a single person must take can lead to careless mistakes. Additionally, decision-makers may lack information, for instance, a manager may not fully understand the implications of certain access privileges. Finally, revoking access privileges can reduce productivity, making it easier for decision-makers to approve assignments rather than to reject them. These challenges, particularly concerning compliance with the Principle of Least Privilege, lead to the following sub-research question:

RQ 2: *How to improve the decision quality in ARs?*

[6]For example, SOX, BSI Grundschutz, or BASEL explicitly demand adherence to the Principle of Least Privilege (it should also be noted that these standards also require the use of SoD).

[7]The term entities in context to IAM refers to employees, permissions or accounts or ACPs (e.g. roles, SoD policies etc.) within this dissertation.

To address RQ2 in this dissertation, the decisions of the ARs are analysed ex-post, i.e. after the execution of an AR campaign *(RC 2)*. The advantage of an ex-post analysis is, that all decisions that were made during the campaign are already available. This provides the opportunity for a comprehensive analysis that can detect structural or organizational issues, which in turn contribute to the sustainable improvement of AR campaigns. It should be noted that RQ 2 could also be tackled by enhancing the review process itself. For example, UI enhancements such as matrix views or dynamically providing additional information to the decision makers represent excellent opportunities in this regard. However, approaches to improve ARs during execution are already addressed in existing literature [16, 3].

Compliance Focus 2: Sustainability of SoD policies

SoD policies ensure that critical processes cannot be executed by a single person. Therefore, they provide security against fraud and insider misuse but also protect employees from accidentally committing illegal actions. Like the Principle of Least Privilege and ARs, SoD policies are a common requirement of most compliance frameworks. In order to comply with SoD requirements in IAM, the following sub-research question arises:

> **RQ 3:** *How to achieve sustainable compliance with SoD requirements in IAM?*

While ARs have not been comprehensively covered in the literature yet, initial research in the area of SoD began in 1987 with the Clark-Wilson model[5]. Since then, there has been a steady stream of publications in the area up to now. Nevertheless, a structured summary of this literature is still lacking *(RC 3.1)*. When addressing RC 3.1 in the context of this work, the results have shown that there is already a significant amount of research regarding specifying, analyzing, and enforcing SoD policies. However, the challenge of deriving and maintaining SoD rules within an organization has not been addressed yet *(RC 3.2)*. While the origin of SoD policies are high-level compliance standards, which often consist of free-flowing text, they must be implemented as concrete rules in IAM that specify explicit exclusions of access privileges. In addition, these rules must be continuously adapted to the changing environment of IAM and changing compliance standards.

3 Research Methodology

This dissertation is grounded in the field of IS research, which is dominated by the research paradigms Design Science and Behavioral Science. Originating from natural sciences, Behavioral Science in IS studies already existing phenomena in order to formulate and evaluate theories that describe or predict human or organizational behavior [14]. Design Science has its roots in engineering and follows a problem-oriented approach by designing, creating and evaluating new artifacts [14]. The two approaches are generally considered to be diametrically opposed in terms of their practical relevance and research

rigor. While Design Science Research[8], with its problem and solution-oriented approach, inherently offers practical utility, the research rigor regarding design, construction, and validation diverges [29]. Conversely, Behavioral Science Research demonstrates research rigor using metrics such as statistical significance. The practical relevance of the results, however, is not always undisputed [29]. Hevner et al.[13] argue that the two approaches are inseparable and draw from each other. The theories of Behavioral Science Research provide a foundation (a *'truth'*) for the artifact development in Design Science Research. In turn, the created artifacts are used to evaluate and verify theories. Therefore, relevance and rigor of research results should be valued equally.

Examining the research questions in Chapter 2, it becomes apparent that this dissertation has to adopt a problem-oriented approach. Therefore, Design Research following the Design Science paradigm is applied as research methodology for this dissertation [9].

Design Science guidelines for IS research

This work consists of several scientific contributions and thus several artifacts were developed and evaluated. Therefore, there is no solid research method valid for the entire work. For example, P3 adheres closely to the Action Design Research method [25], a popular approach in Design Science Research [6], while others adopt varying aspects of the paradigm. Hevner et al. [14] propose seven research guidelines for Design Science in IS that were applied in the publications of this work. They emphasize that the guidelines should not be applied compulsively or routinely. Depending on the context, scientists should decide for themselves how and to what extent the guidelines are adhered to.

Guideline 1: Design as an Artifact

The result of Design Science Research should be an IT-artifact, which addresses a crucial organization problem. The artifact can be (according to [14]) a construct, a model, a method, or an instantiation. This is one of the fundamental differences to Behavioral Science Research, as Design Science Research deals with construction of an artifact rather than studying existing phenomena. In this dissertation, the results of the publications are IT-artifacts.

Guideline 2: Problem Relevance

Design Science Research should develop technology-based solutions/ IT-artifacts for unsolved and important business problems. Therefore, the relevance of the research depends on its utility for practitioners in the field of IS. The overall research question

[8] The term Design Research refers to the development of artifacts for general problems in IS. Design Science is the 'meta-research' that explores the process of how Design Research is done. Finally, Design Science Research comprises research that can be associated with the Design Science paradigm, i.e. both Design Research and Design Science. [29]

[9] In P2 and P4 systematic literature reviews were carried out. Therefore appropriate methods and approaches for systematic literature reviews were applied that cannot be directly assigned to the Design Science paradigm (e.g. [28, 22], among others).

posed in Chapter 2 tackles an important real-world problem relevant to highly regulated and medium- to large-scale organizations: How to achieve sustainable compliance in IAM. Additionally, in several publications practitioners in the field of IS and IAM were involved in problem definition and artifact development.

Guideline 3: Design Evaluation

The developed IT-artifacts have to undergo rigorous evaluation for their quality, utility, and efficacy. Applicable evaluation methods are observational, analytical, experimental, testing, and descriptive. All IT-artifacts developed in this dissertation were properly evaluated in collaboration with large and highly regulated organizations. These partner organizations provided real-world data as well as feedback on the research results.

Guideline 4: Research Contributions

The result of Design Science Research should be a clear contribution, which takes the form of an IT-artifact, design construction knowledge or design evaluation knowledge. Artifacts generate significant value for research when they are applied, examined, and evaluated in a suitable environment [14]. The research contributions of this work are IT-artifacts that have been applied to real-world problems and are studied using real-world data. This ensures an appropriate contribution to research.

Guideline 5: Research Rigor

Design Science Research should rigorously utilize methods for IT-artifact construction and evaluation. However, the rigor should be in relation with the practical utility of the IT-artifact. In other words, "blind" adherence to research methods without considering practical applicability can reduce utility and relevance of the research. Within this work, research rigor is ensured by adhering to these guidelines and utilising methods such as proposed by Sein et al.[25].

Guideline 6: Design as a Search Process

When designing an IT-artifact to solve a research problem, there is not always an optimal solution. Therefore, an iterative approach is recommended. It seeks a heuristic solution by alternately generating an IT-artifact and then evaluating it against the requirements. The results of the evaluation are then considered in the next construction phase. This iterative approach was applied to the artifact development in this work. In addition, real-world data and feedback from practitioners could often be used for the evaluation.

Guideline 7: Communication of Research

Finally, Design Science Research should be communicated in a way that is understandable for both technology-oriented and management-oriented audiences. This is achieved in

this work by publishing the research results in scientific conferences and journals. The individual publications have high-level descriptions that are interesting to management. These can be found in the abstracts and introductions. Each IT-artifact is also described in detail so that technology-oriented audiences can understand and implement it. In addition, the journals and conferences are from the field of IS and have been selected in such a way that they are of interest to both audiences.

4 Results

In Chapter 2, three research areas for the dissertation are derived from the primary research question, each with its own sub-research question (RQ1-RQ3). Research challenges (*RC 1.1*, *RC 1.2*, *RC 1.3*, *RC 2*, *RC 3.1*, *RC 3.2*) are derived from the sub-research questions, which are addressed in the publications of this dissertation. This chapter provides an overview of these publications and shows how they fit into the overall concept of the dissertation. In particular, it will be discussed how the publications address the research challenges.

4.1 Overview of the Research Contributions

Table 1 lists the publications that result from this dissertation. The column "Type" shows whether a publication is a conference (C) or a journal (J) paper. The "Status" column indicates whether the publication has already been published (pub.) or just submitted (sub.). Furthermore, the table provides an overview of the research challenges addressed by each respective publication. Note that P4 has been submitted and is currently undergoing revision in the submission process.

The overall framework of the dissertation, the publications and the addressed research challenges are depicted in Figure 1. The first research area ("*Foundation*") forms the basis of this dissertation and includes publications P1 and P2. P1 introduces a generic conceptual IAM model (*RC 1.1*) derived from multiple IAM standards and thus shows broad applicability. This conceptual IAM model is a central element and is reused and extended in P2, P3 and P5. P1 also addresses *RC 1.3*, by proposing Total Attribute Quality Management (TAQM), a process-oriented approach for assessing and enhancing IAM attribute data quality. Consequently, TAQM also builds on the conceptual IAM model. P2 tackles *RC 1.2* by sharpening the term *quality* in the context of access control policies. More precisely, six relevant quality criteria for access control policies are derived by analysing existing literature. P2 also contributes to *RC 1.3* by conducting a structured literature survey that identifies papers addressing the improvement of the six identified quality criteria.

Compliance Focus 1 addresses ARs and contains contribution P3. P3 proposes an approach to assess the quality of AR decisions after the execution of an AR campaign (*RC 2*). Identifying wrong decisions is one way to improve the quality of ARs, as it may

No.	Publication	Type	Status	RC
P1	Michael Kunz, Alexander Puchta, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications (JISA)*, 44:64–79, 2019.	J	pub.	*RC1.1*, *RC1.3*
P2	Sascha Kern, Thomas Baumer, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Optimization of access control policies. *Journal of Information Security and Applications (JISA)*, 70:103301, 2022.	J	pub.	*RC1.2*, *RC1.3*
P3	Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. Monitoring access reviews by crowd labelling. In Simone Fischer-Hübner, Costas Lambrinoudakis, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Trust, Privacy and Security in Digital Business (TrustBus)*, pages 3–17, Cham, 2021. Springer International Publishing.	C	pub.	*RC 2*
P4	Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Separation of duty in information security. *ACM Computing Surveys*.	J	sub.	*RC 3.1*
P5	Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. A framework for managing separation of duty policies. *In Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA, 2024. Association for Computing Machine.	C	pub.	*RC 3.2*

Table 1: Overview of the publications within this dissertation

indicate structural or organisational problems in AR campaigns. Therefore, P3 uses the conceptual IAM model developed in P1 to formalize AR campaigns.

Compliance Focus 2 deals with sustainability of SoD policies and includes the publications P4 and P5. P4 is an exhaustive, structured literature survey that sums up publications in the area of SoD published between 1990 and 2023 (*RC 3.1*). The survey reveals that there is an extent of literature on the specification, enforcement, and analysis of SoD rules. However, a suitable approach for deriving and maintaining SoD rules within organizations is currently still lacking. This challenge is addressed in P5, which proposes a management framework for SoD policies, which especially considers the various stakeholders involved in the derivation and maintenance processes (*RC 3.2*). One of the central challenges is that knowledge about SoD rules is spread across the organization. Once again, the conceptual IAM model of P1 is used and extended to provide a solid understanding of the IAM-related entities. In the following, the publications are presented

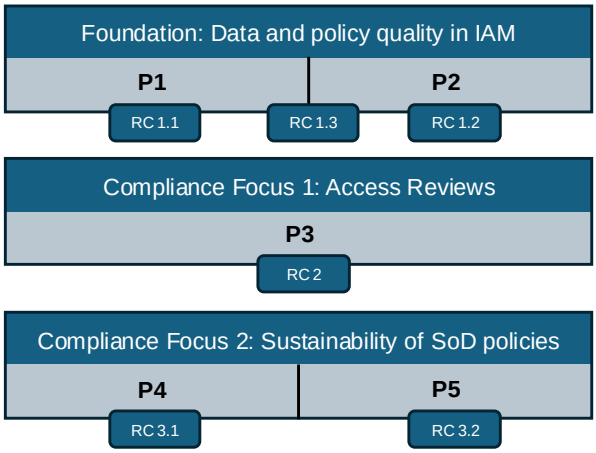


Figure 1: Classification of publications and research challenges in the context of this work

in more detail, especially in the context of this dissertation and with regard to the research challenges.

4.2 Foundation: Data and policy quality in IAM

The first research area addresses RQ 1 and deals with the challenge of defining and ensuring ACP and attribute data quality in IAM. Data quality is the foundation for ensuring seamless operation in IAM. On the one hand it is crucial for adhering to the Principle of Least Privilege and on the other hand the success of other compliance measures relies on it.

P1: Attribute Quality Management for Dynamic Identity and Access Management

The central entities in IAM are characterized by a variety of attributes. For example, employees may be assigned to a function (e.g. "Developer", "Consultant"), an email address or a responsible manager. The goal of P1 is to develop a comprehensive approach for assessing, maintaining and enhancing the quality of those attributes (*RC 1.3*). In order to provide a general basis for establishing IAM quality requirements that can be used to compare existing quality management approaches, a general conceptual IAM model was developed (*RC 1.1*). The model was created by examining several relevant IAM standards and extracting the most important entities as well as their relationships with each other.

Figure 2 shows the conceptual IAM model and is roughly explained below as it also provides a foundation for P2, P3 and P5. Employees (= digital identity[10] in Figure 2) can have several accounts and can be member of one or more business roles. The employees are assigned to permissions[11] via their accounts or indirectly via business roles. The

[10]The term *digital identity* originates from Pfitzmann and Hansen [23] and describes a set of attribute values which are accessible by technical means and belong to/ identify an individual person. In the context of enterprise IAM, *digital identities* are the digital representations of the organizations' employees (e.g., in human resource systems). For the sake of simplicity, the term *employee* is used instead of *digital identity* in this work. Note, that *employee* does not refer to the person directly, but to their digital representation.

[11]The term permission and access right can be used interchangeably in this work.

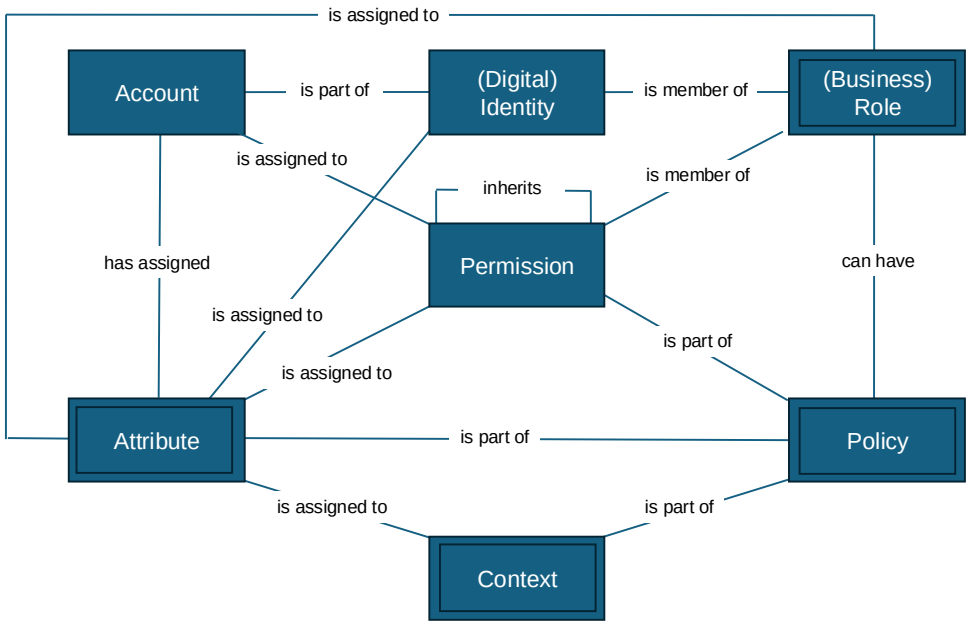


Figure 2: Generic conceptual IAM model according to Kunz et al. [18]

four core entities (i.e. employees, accounts, business roles, access rights) have assigned attribute values. Policies are rules that determine, based on attribute values, which access rights and business roles are assigned to employees and accounts. In P1, the term *policy* refers to the concept of ABAC policies, a more general interpretation also allows the instantiation of SoD rules that contain mutual exclusive permissions and business roles.

After creating the conceptual IAM model, we established six quality criteria that define the specific IAM requirements for data quality management approaches. One criterion is the capability to integrate the entities of the conceptual IAM model. Twelve data quality management approaches were identified and ranked according to the defined quality criteria. However, none of them could fulfil all the criteria. Consequently, a new approach was developed which builds upon the three approaches that best meet the defined IAM quality criteria (i.e. the upper quartile). The proposed TAQM (Total Attribute Quality Management) approach is specifically optimized for assessing and improving IAM attribute quality in IAM environments. TAQM is an iterative / cyclic approach comprising four high-level phases as well as detailed guidelines for the low-level implementation of these phases. TAQM is a hybrid approach that supports the integration of human experts as well as (semi-)automated tools. The four phases of TAQM are depicted in Figure 3 and shortly described below.

In Phase I the relevant attributes for the approach are selected. This is done by human experts with the support of tools such as data browsing interfaces or grouping and analytic functionalities. In Phase II, the quality of the selected attributes is assessed using tool-based techniques optimized for entities of the conceptual IAM model. In general, TAQM allows flexibility in the assessment of attribute data quality, for example it could be estimated by attribute owners. However, a semi-automatic approach is proposed

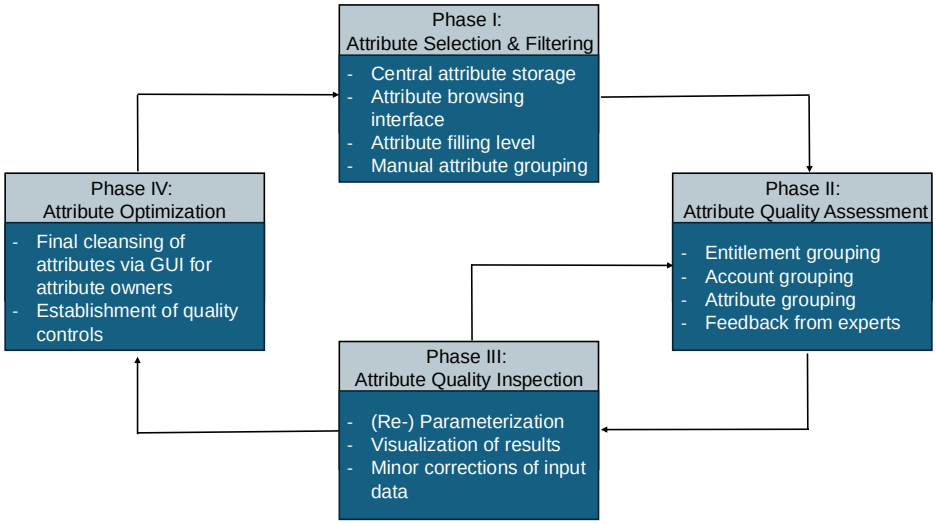


Figure 3: Phases of the TAQM approach according to Kunz et al. [18]

and recommended, which assumes an interdependence between the different attribute values. Phase II creates findings, i.e. possible wrong attribute values, that have to be manually inspected by human experts in Phase III. Again, a tool-supported approach utilizing visualizations is suggested. It is also possible to redefine parameters and rerun Phase II to improve the results. Finally, in Phase IV the erroneous attributes found are corrected. However, Phase IV reaches even further and searches for structural causes of the identified errors, which can then be addressed in the long term through organizational and technical measures. P1 also evaluates the TAQM approach by applying it to three real-world use cases from different organizations.

Contribution of P1

P1 addresses *RC 1.1* by developing a generic conceptual IAM model that comprises the central entities in IAM and their relations with each other. P1 additionally addresses *RC 1.3* by proposing TAQM, an iterative approach for assessing and improving IAM attribute quality.

P2: Optimization of Access Control Policies

After ACPs (e.g., ABAC policies, roles, SoD rules, etc.) are modeled, it is important to keep them at a high quality. First, P2 demonstrates that ACP quality is an instance of the quality concept proposed by Wand and Wang [27]. Based on this, the quality of ACPs can be defined *"as their fitness for use with regard to one or more quality dimensions that reflect the application context of access control."*[17] This definition is completed by providing a list of sixteen quality dimensions (cf. Table 2). These quality dimensions are derived from the existing literature, but it should be noted that they are not intended to be exhaustive. The proposed definition tackles *RC 1.2*.

Quality dimension	Survey	Quality dimension	Survey
Accuracy		Conflicts	x
Excessive UPAs	x	Grade of automation	x
Missing UPAs	x	Evaluation runtime	
Maintainability		Similarity to opt. state	
Understandability		Risk	
Semantic meaningfulness		Completeness	
Complexity	x	Usage	
Redundancy	x	Relevance	

Table 2: The 16 ACP quality dimensions proposed by Kern et al.[17] and their use in the survey

Based on this definition, a structured literature survey is conducted to find current work dealing with the improvement and optimization of ACP quality (the identified literature addresses *RC 1.3*). Therefore, six of the sixteen quality dimensions (cf. column 'survey' in Table 2) are selected in order to limit the scope of the survey to a reasonable size. The survey resulted in 61 publications, 42 of which deal with role optimization (RBAC), 21 deal with ABAC policy optimization and two deal with both RBAC and ABAC. The analysis of the identified publications provided several core insights:

1. **Identification of user-permission assignment (UPA)[12] inaccuracies**
   Dealing with the identification of UPA inaccuracies (i.e. missing UPAs, excessive UPAs) is one of the most important objectives in improving the quality of ACPs. It ensures productivity and adherence to the Principle of Least Privilege. Analysed approaches deal with manual identification (e.g. ARs), the identification based on usage statistics and the identification based on the update or change history of the entities.

2. **Data Availability**
   A core requirement for improving ACP quality is an integrated view of the required IAM data (cfg. TAQM from P1). This includes the availability of the attributes of the entities, the access logs and the update logs. In particular, access logs are difficult to obtain for technical, organizational and ethical reasons [17].

3. **Minimal Perturbation**
   When optimizing ACP quality, it is preferable to make as few changes as possible. On the one hand, changes generate organizational effort and, on the other hand, too many changes corrupt the original structure of the ACPs. Usually the original structure carries semantic meaning that should be retained. For instance, business roles may be associated with organizational functions and contain respective permissions.

[12] In this work, a UPA refers to the *actual* assignment of a permission to an employee after evaluating the ACPs. Considering the conceptual IAM model for example, it does not matter whether the employee is granted a permission via (business) role or via an account. The only relevant point is that, in the end, the employee is assigned to the permission.

4. **Recommendation-Based Optimization**
   The results from automated improvement and optimization methods should not be applied automatically to the ACPs. A hybrid approach consisting of automated suggestions and a human decision maker is recommended (cfg. TAQM from P1). Human deciders can prevent erroneous changes and ensure that regulatory requirements are met. In addition, the semantic meaning of ACPs can be retained. In many cases, a human decision maker is also required to ensure compliance with approval processes in which a responsible person has to approve the changes.

Some of these points are also considered in P1, P3 and P5 (although P1 and P3 were published earlier). For example, P1 and P3 propose hybrid recommendation-based approaches. P1 recommends a central attribute storage and P5 aims to preserve the semantic meaning of SoD policies by splitting policy maintenance into an organizational and a technical layer. The approach seeks to simplify and reduce changes at the organizational layer.

Contribution of P2

P2 addresses *RC 1.2* by providing a definition of the term ACP quality as well as sixteen quality dimensions that finalize this definition. Additionally, *RC 1.3* is addressed by identifying and analyzing literature that deals with the improvement and optimization of ACP quality.

4.3 Compliance Focus 1: Access Reviews

Compliance Focus 1 tackles RQ 2 and therefore deals with the issue of assessing and improving the decision quality in ARs. More precisely, *RC 2* is addressed in P3, which deals with the ex-post analyses of AR campaigns.

P3: Monitoring Access Reviews by Crowd Labelling

As mentioned earlier in this work, the execution of ARs is very time-consuming because all assignments have to be inspected manually. This inevitably leads to wrong decisions, e.g. due to careless mistakes or because someone has to make a lot of decisions. P3 aims to detect those wrong decisions after an AR campaign was executed. This can help to uncover structural problems of AR campaigns, e.g. insufficient working instructions or overburdened decision makers.

In a first step the problem of ARs is generalized using the conceptual IAM model of Kunz et al.[18]. It is shown that the problem of assessing the quality of AR decisions can be seen as an instance of assessing crowd labelling decision quality. In crowd labelling, several individuals join to perform a labelling task together. This can also lead to loose or public collaborations, for example anyone can join the Galaxy Zoo[13] project and help to label universes. ARs can also be seen as labelling tasks, where multiple decision

[13] https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/

makers join together and label IAM-related assignments. However, there are some aspects making ARs a special case of crowd labelling:

- **Binary decisions:** While crowd labelling projects like Galaxy Zoo usually have multiple label values, ARs allow two label values ('keep relation', 'remove relation').
- **Sole deciders:** Unlike most crowd labelling projects, each decision in an AR is made by one single decision maker.
- **Principle of Least Privilege:** From a security perspective, assessing the quality of positive decisions in ARs is more important then assessing the quality of decisions in which an assignment is removed.

Naturally, crowd labelling suffers from phenomena such as spammers and therefore has to deal with wrong decisions just like ARs. Thus, a literature search was conducted and a machine-learning based approach [9] for assessing crowd labelling decisions could be identified that considers the specific requirements of ARs. In this approach, a model is generated and trained for each decision maker, which can then evaluate the decisions of the other decision makers.

Adopting the approach to ARs and evaluating it in a real-world IAM project of an industry partner initially yielded insufficient results: every model suggested keeping all assignments. Analysis of these results led to further insights into the peculiarities of ARs as an instance of the crowd labelling problem:

- **Many positive decisions:** Since it can be assumed that most assignments of the existing ACPs are correct, the frequently chosen decision in ARs is 'keep relation' (93.9% in the provided real-world IAM data).
- **Unqualified models:** Models trained on data unrelated to the decision being evaluated will make incorrect decisions. For example, a model that has been trained on decisions that affect external HR employees is not qualified to evaluate decisions for internal IT employees.
- **Noise:** In the real-world data there are several deciders with very few decisions. Models based on these deciders evaluate decisions without suitable foundation.

Consequently, the initial approach [9] was further improved to address these challenges. For each decision a model evaluates, an Experience Score is calculated. This Experience Score becomes higher the more similar the decisions of the trained model are to the decision to be evaluated. In the next step, models whose Experience Score falls below a certain threshold in the evaluation of a decision are removed. The suggestions of the remaining models are then weighted with their Experience Score. In the subsequent evaluation, these changes led to good results for the real-world data. The incorrect decisions found could also be verified with the industry partner.

P3 has some valuable insights for further research: On the one hand, it is shown that ARs are an instance of crowd labelling. This allows future research to apply crowd labelling approaches to ARs. On the other hand, an existing crowd labelling approach is further improved. Note that the structure of P3 strictly follows the ADR method. For a detailed description of the ADR method execution, please see P3 [11].

Contribution of P3

P3 addresses *RC 2* by proposing an approach for assessing the quality of crowd labelling decisions ex post. Furthermore, it is demonstrated that ARs are an instance of crowd labelling, which allows future research to use crowd labelling approaches on ARs.

4.4 Compliance Focus 2: Sustainability of SoD policies

Compliance Focus 2 addresses RQ 3 and deals with the question of how to achieve sustainable SoD compliance in IAM. The current state of research is first surveyed in P4 and then an approach for creating and maintaining SoD rules in an organizational context is proposed in P5.

P4: Separation of Duty in Information Security

SoD in IS is a multi-faceted subject. It involves both organizational and technical challenges and is explored in several research areas. For instance, SoD is addressed in research related to IAM, modelling languages, workflows, software development, databases and many others. However, no comprehensive survey regarding SoD in IS has been conducted so far.

Thus, P4 aims to close this research gap by providing an exhaustive survey of the relevant literature. This is done by addressing three research questions with the following objectives:

- Identify the relevant literature in SoD
- Structure and categorize this relevant literature
- Analyze the identified literature and determine current challenges and the state of the art

To identify the relevant literature in SoD a structured standalone literature review was conducted. The Eight-Step Guide of Okoli and Schabram [22] was utilized as an overall process model along with several other methodical guidelines. The literature search yielded 3,171 results after query execution, 88 publications of which were identified as relevant for further analysis.

After identifying the relevant publications a multi-layer classification scheme for SoD literature is proposed. The classification scheme consists of 26 classes distributed over

four layers. The primary categories, i.e. the categories of classification layer 1, are briefly explained below:

- **SoD Specification** *(39 publications)*: Publications dealing with the specification of SoD address the question of how SoD can be expressed in the various models and approaches, such as RBAC, ABAC, workflows, petri-nets etc.. This also includes the extension of existing approaches as well as the issue of how SoD rules can be structured and expressed in general.

- **SoD Enforcement** *(26 publications)*: After the rules have been specified, they must also be enforced. Research relevant to this category tries to identify existing SoD violations or prevent them in the first place. For instance, SoD enforcement can be studied in software development or it can be examined under specific circumstances, such as in the context of distributed systems or cloud computing.

- **SoD Analyses** *(17 publications)*: This category contains publications that propose algorithmic approaches for evaluating, analyzing, simulating or transforming SoD policies. For instance, role-mining, i.e. the algorithmic detection of roles, can be conducted with respect to SoD rules.

- **High-level SoD & Management** *(6 publications)*: While most publications deal with SoD on a 'lower' or technical level, there are some publications that address organizational aspects of SoD. Examples include the implementation of SoD within organizations or the introduction of maturity models for SoD.

For the full classification scheme as well as a detailed explanation of the categories and of each publication, please see P4 [10].

Finally, to address the last objective, the identified publications are analyzed. Through this analysis, three major trends are identified, which are regularly addressed in the publications: Access Control Models, Workflows, and High-level SoD & management. In addition, four research areas have been identified, which require further investigation and analysis:

- **Human understandability:** The comprehensibility of SoD policies plays an important role in ensuring quality and ease of maintenance. For example, visualizations could make the policies more accessible and easier for people to understand.

- **Creation of SoD rules:** A related issue is that organizations must first establish their SoD rules. This should be done on basis of existing laws and regulations. While most research assumes that these rules already exist, only one paper could be identified in the conducted literature review that addresses this topic.

- **Social aspects of SoD:** Compliance with SoD policies is ensured - among other things - by employees overseeing the actions of each other. However, employees may try to avoid conflict with colleagues or risk their career options if they report a

superior. These social aspects of SoD have received little attention in research so far.

- **ABAC:** While ABAC is seen as the successor of RBAC, only little SoD research (4 publications) addresses ABAC.

Please note that neither the analyzed major trends nor the identified research gaps are meant to be exhaustive. Overall, it can be said that one central insight of P4 is that the organizational context of SoD should be addressed more often in studies to achieve more practical relevance in research.

At the time of writing this work, P4 has been submitted to the Journal *ACM Computing Surveys* and is currently under major revision.

Contribution of P4

P4 addresses *RC 3.1* by surveying, categorizing and analyzing the existing literature in SoD dating back to 1990. In addition, major trends and research gaps are identified.

P5: A Framework for Managing Separation of Duty Policies

In P4 several research gaps have been identified in SoD research. The topics "human understandability" and "creation of SoD rules" are addressed in P5. More precisely, a framework for creating and maintaining SoD rules that focuses on human understandability and shared knowledge is proposed. For this purpose, semi-structured interviews were conducted with a total of 6 organizations employing 300 to 33,000 employees in the engineering, telecommunications, finance, and insurance sectors. The primary goals of the interviews were to understand how the organizations structure their SoD policies, how they create understandability, which people are involved and what the processes for creation and maintenance look like.

In the interviews three prototypical types of stakeholders (i.e. responsible persons) for SoD rule creation and maintenance were described:

- **IAM Governance:** Responsible for complying with laws on a high level. Understands the laws and legal requirements, but has no deeper technical understanding.
- **IAM Team:** Responsible for IAM operations and for IAM projects, also responsible for SoD adherence on a technical level, provides infrastructure and manages SoD creation, maintenance and enforcement.
- **Domain Experts:** Technical or business experts who are familiar with users and permissions. They do not have a deeper understanding of laws and regulations.

The stakeholders and their responsibilities are visualized in Figure 4. For a more detailed description, please see P5 [12]. In addition, three primary ways to define SoD rules in organizations were identified based on the interviews:

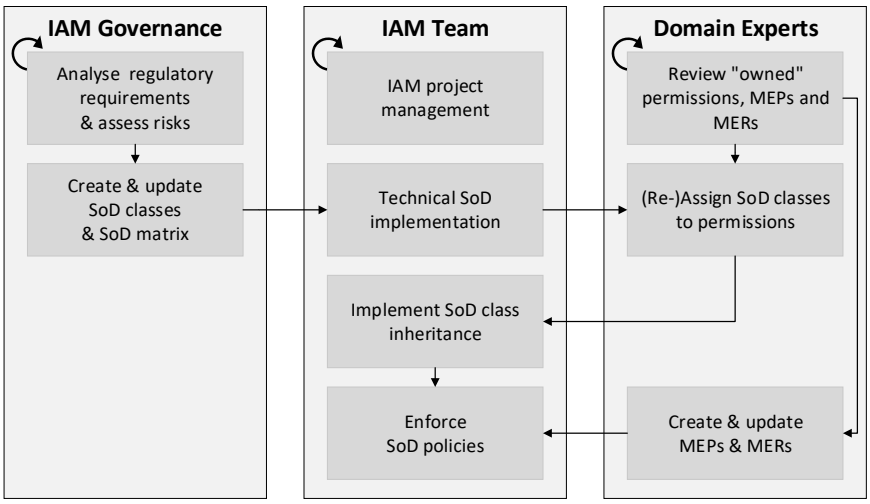


Figure 4: Stakeholders and responsibilities according to the approach of P5[12].

- **Buying rule sets:** For certain systems, accounting and consulting firms have already designed predefined SoD rule sets that adhere to specific compliance standards. While buying SoD rules can be a sufficient solution, SoD rule sets for purchase are not available for all applications.
- **Individual fine-grained rules:** There are individual permissions and roles that are mutually exclusive to each other. Specifying those mutually exclusive roles (MER) and permissions (MEP) requires a deep understanding of processes and applications and is carried out by Domain Experts.
- **SoD Classes and SoD Matrix:** SoD Classes are defined by IAM Governance and assigned to permissions by Domain Experts. Mutually exclusive SoD Classes are defined in a so-called SoD Matrix. This is described in more detail below.

The SoD Matrix and SoD Classes simplify the creation and maintenance of SoD rules by distributing the task to different stakeholders based on their knowledge: In a first step, IAM Governance analyzes the existing rules and laws. Based on this, SoD Classes and an SoD Matrix are defined. SoD Classes can be defined as "groupings of tasks, functions or responsibilities in an organization that are designed to conflict with each other." [12] The names of SoD Classes already carry a meaning like "Market" or "Market follow up", additionally they can have a description. Mutually exclusive SoD Classes are defined in the SoD Matrix. The SoD Classes are assigned to the permission of the organizations by the Domain Experts. The Domain Experts understand the purpose of the respective SoD Class and know the function of the permissions. They can therefore assign the SoD Classes to the permissions without knowing the underlying laws and guidelines or the SoD Matrix. Based on the assigned SoD Classes and the SoD Matrix, exclusive permissions are determined. For various reasons, there may still be permissions that are mutually exclusive, even if they do not conflict in the SoD Matrix. For this purpose,

individual fine-grained rules (MERs and MEPs) can be defined (which is usually done by Domain Experts).

The approach adds purpose to SoD rules, as they are defined by the SoD Matrix, which in turn is created by IAM Governance through interpretation of laws and regulations. Maintenance is also simplified: if permissions or applications change, only the assignments of SoD Classes to permissions need to be changed by the Domain Experts. When regulatory requirements or laws change, only the SoD Matrix needs to be updated by IAM Governance.

To streamline this framework with current research, the conceptual IAM Model is extended and SoD Classes, SoD Matrix as well as MERs and MEPs are added as entities (cfg. Figure 5).

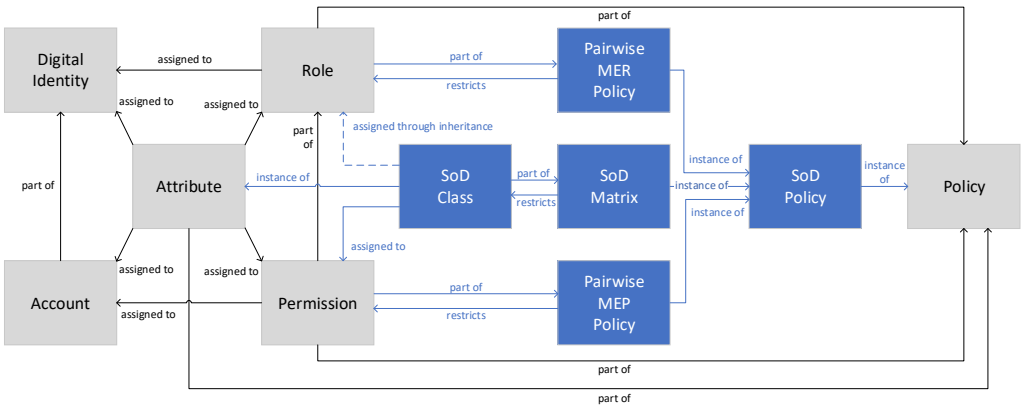


Figure 5: Extension of the conceptual IAM model according to [12].

To further demonstrate the applicability in existing research approaches, the proposed framework is formalized within the RBAC model. On a theoretical level, it is shown that the corresponding policies are enforceable. In addition, to demonstrate the practical applicability, the proposed framework is evaluated with real-world data provided by a large financial services provider that has already implemented SoD Classes and a SoD Matrix. The technical applicability of the approach as well as the simplification for policy management are demonstrated by showing that 12.295 SoD rules (i.e. conflicting permissions) can be expressed with 14 SoD Classes and 32 SoD Matrix entries in the real-world data.

Contribution of P5

P5 addresses *RC 3.2* by proposing an approach for deriving and maintaining SoD rules within organizations. Particular attention is given to comprehensibility and the distribution of knowledge within companies.

4.5 Complementary Publications

In addition to P1-P5, further complementary publications were issued related to this dissertation. Although these works do not directly belong to this dissertation, they have

nevertheless influenced it and should be mentioned for the sake of completeness. Table 3 summarizes these publications. The columns are analogous to those in Table 1. The publications are briefly summarized below.

No.	Publication	Type	Status
A1	Matthias Hummer, Sebastian Groll, Michael Kunz, Ludwig Fuchs, and Günther Pernul. Measuring identity and access management performance - an expert survey on possible performance indicators. In Paolo Mori, Steven Furnell, and Olivier Camp, editors, *Proceedings of the 4th International Conference on Information Systems Security and Privacy, ICISSP 2018, Funchal, Madeira - Portugal, January 22-24, 2018,* pages 233–240. SciTePress, 2018.	C	pub.
A2	Markus Hornsteiner, Sebastian Groll, and Alexander Puchta. Towards a user-centric iam entitlement shop - learnings from the e-commerce. In *13th International Conference on Security of Information and Networks*, SIN 2020, New York, NY, USA, 2021. Association for Computing Machinery.	C	pub.
A3	Alexander Puchta, Sebastian Groll, and Günther Pernul. Leveraging dynamic information for identity and access management: An extension of current enterprise iam architecture. In *International Conference on Information Systems Security and Privacy*, 2021.	C	pub.

Table 3: Overview of the complementary publications related with this dissertation

Publication **A1** takes first steps towards sustainable IAM maintenance by developing performance indicators for IAM. Therefore, the Goal-Question-Metric paradigm[2] is applied in order to derive possibly relevant performance indicators. Afterwards a survey with 32 participants specialized in IAM was carried out to evaluate the derived performance indicators. Reducing risk, improving data and process quality, and meeting compliance requirements could be identified as the most important goal in the study.

Publication **A2** is a paper that studies IAM entitlement stores. These are request portals where permissions and other functions can be ordered, similar to an online store (e.g. Amazon[14]). At first a survey is conducted in order to work out the shortcomings of current IAM entitlement stores. The shortcomings identified include the difficulty in finding the correct permissions, the lack of information about permissions, and the lack of transparency in the ordering process. To address these shortcomings, a number of improvements that can be learned from the e-commerce (e.g., Amazon) is proposed. For example, the status of a permission order could be tracked like the delivery of a package. However, since **A2** is a short paper, a major drawback is that the proposed improvements have not been implemented or evaluated.

Publication **A3** addresses the challenge that IAM relevant information is distributed across many different applications within an organization (e.g. user statistics or HR information). To tackle this issue an InterConnectivity Module (ICM) is proposed as an

[14]https://www.amazon.com/

intermediary between other applications and the IAM system. The ICM can prepare and export the data to the IAM system by carrying out four steps (Data Harmonization & Quality Monitoring, IAM Data Analysis, Result Verification and Data and Result Storage). Basic IAM functionalities such as provisioning are still carried out via directly connecting the IAM system to the applications. The approach was evaluated in two real-life use cases by enhancing the IAM data with the additional information and performing ARs both with and without the additional data. It was shown that significantly more permissions were revoked in ARs enhanced with the additional information from the ICM.

5 Conclusion and Future Work

IAM plays a central role in adhering to compliance regulations regarding IS security. However, due to the fluctuating nature of IAM and IAM-relevant entities, implementing compliance measures in a sustainable and maintainable way becomes a major challenge. Thus, the goal of this dissertation is to sustainably improve compliance and compliance measures in the context of enterprise IAM. For this purpose, the dissertation is divided into three parts: Data and policy quality in IAM and two central IAM-related compliance measures: ARs and SoD.

As a foundation, this work aims to sustainably improve the quality of data and policies within IAM. High IAM data quality is required for a functioning IAM and for compliance with the Principle of Least Privilege, which is demanded by many compliance and security standards. In addition, data quality is a prerequisite for all other compliance measures e.g. ARs(cf. P3) and SoDs (cf. P5). This is actually a symbiotic relationship: compliance measures, among other goals, seek to improve data quality, but their effectiveness also depends on the quality of the data.

As a basis for data quality improvement, but also as a basic data model for further research, a generic conceptual IAM model is derived by combining several relevant IAM standards. Additionally, the TAQM approach is proposed, which is an iterative approach for evaluating and improving the quality of IAM attributes. Furthermore, the dissertation clearly defines the term ACP quality and derives sixteen relevant quality dimensions. In order to identify approaches that deal with the improvement of ACP quality, the relevant literature for six selected quality dimensions is survey and analyzed. After providing the foundation for further compliance-related research and addressing IAM attribute quality and ACP quality, the dissertation focuses on enhancing two central IAM-related compliance measures: ARs and SoDs.

ARs have not yet been covered in detail in the literature, but there are some approaches that address the execution of ARs, i.e. the activity in which the reviewers make their decisions. Within this dissertation, an approach is proposed that assesses the decision quality of ARs ex-post. Thus, all decisions made in the AR campaign can be taken into account and structural problems of the AR campaign can be discovered (e.g. overworked reviewers). In addition, it is shown that ARs are a special case of crowd labelling and a

formalism connecting the two research realms is proposed. This provides a promising foundation for future research.

SoD has been the subject of research for decades, but a comprehensive survey on the subject is still lacking. To close this research gap, a rigorous and structured literature survey, summarizing relevant work in the research realm dating back to 1990, was conducted. A categorization for SoD literature was developed and major trends as well as current research gaps were identified. Consequently, the research gaps which were considered to be of particular importance and practical relevance ("Creation of SoD rules", "Human understandability") are addressed. Because these issues are already addressed in practice, interviews were first conducted with experts from large companies. Based on these interviews and building on the conceptual IAM model, an approach for creating and maintaining SoD rules was developed that particularly focuses on human understandability. The extension of the conceptual IAM model builds a bridge to science and ensures compatibility with other approaches from literature. It is also shown prototypically how the approach can be formalized and enforced within the RBAC model.

In the course of this work, several starting points for further research have emerged. For IAM in general and specifically for compliance in IAM, it would be beneficial to create opportunities to conduct more practice-oriented research. Established models in research such as RBAC or ABAC are not suitable enough and not intended to reflect how modern IAM systems are used in practice throughout organizations. Working with experts from practice during this dissertation has shown that the conceptual IAM model is more suitable, but the model is only based on IAM standards and frameworks and does not consider real-life IAM systems. A study that brings together modern IAM systems, IAM landscapes in organizations and theoretical approaches could develop IAM models that are closer to practice and thus pave the way for more practical relevance in IAM research .

Regarding ARs, only little research exists, even though it is an important area that is costly and time-consuming for organizations. This dissertation shows that ARs are an instance of crowd labelling, which provides new opportunities for future research, e.g. applying methods from crowd labelling to ARs. In addition the recent developments in artificial intelligence should be considered. For example, during AR execution the user interface could be enhanced using artificial intelligence (e.g. with a chat bot guiding the user, or with smart suggestions, etc.). While companies are taking their first steps in this direction, research should also examine the extent to which artificial intelligence improves the process, which risks exist and which ethical issues might arise.

Future research regarding SoD is already discussed in P4 ("Human understandability", "Creation of SoD rules", "Social aspects of SoD", "SoD in ABAC"). However, the influences of artificial intelligence should also be investigated in this research area. For example, SoD rules could be automatically generated by parsing legal texts or compliance standards. Since artificial intelligence has currently still many problems (e.g.

"hallucinating"), hybrid approaches to quality assurance could be investigated (among many others).

Part II

Research Papers

1 Attribute Quality Management for Dynamic Identity and Access Management

Current status:	Published
Journal:	Journal of Information Security and Applications, Volume 44, February 2019
Date of acceptance:	November 10, 2018
Full citation:	Michael Kunz, Alexander Puchta, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications (JISA)*, 44:64–79, 2019.
Authors contributions:	Michael Kunz 50% Alexander Puchta 15% Sebastian Groll 15% Ludwig Fuchs 10% Günther Pernul 10%

Conference Description: Journal of Information Security and Applications (JISA) focuses on the original research and practice-driven applications with relevance to information security and applications. JISA provides a common linkage between a vibrant scientific and research community and industry professionals by offering a clear view on modern problems and challenges in information security, as well as identifying promising scientific and "best-practice" solutions. JISA issues offer a balance between original research work and innovative industrial approaches by internationally renowned information security experts and researchers.

Attribute Quality Management for Dynamic Identity and Access Management ☆

Michael Kunz[a,*], Alexander Puchta[a], Sebastian Groll[a], Ludwig Fuchs[b], Günther Pernul[a]

[a] *University of Regensburg, Universitätsstr. 31, 93053 Regensburg*
[b] *Nexis GmbH, Franz-Mayer-Str. 1, 93053 Regensburg*

Abstract

Identity and access management (IAM) has become one main challenge for companies over the last decade. Most of the medium-sized and large organizations operate standardized IAM infrastructures in order to comply with regulations and improve the level of IAM automation. A recent trend is the application of attribute-based access control (ABAC) for automatically assigning permissions to employees. The success of ABAC, however, heavily relies on the availability of high-quality attribute definitions and values. Up to now, no structured attribute quality management approach for IAM environments exists. Within this paper, we propose TAQM, a comprehensive approach building on a tool-supported structured process for measuring and improvement of IAM data quality. During the evaluation of three real-life use cases within large industrial companies we underline the applicability of TAQM for the identification and cleansing of attribute errors by IT and non-IT experts as well as the general introduction of quality management processes for IAM.

Keywords: Identity Management, Identity and Access Management, Access Management, Attribute Quality, Quality Management, Attribute-based Access Control, ABAC

1. Introduction

In order to provide secure and compliant access to IT resources, centralized identity and access management (IAM) has become one of the main challenges for companies. The successful fulfillment of existing compliance requirements is one of the core drivers when implementing IAM infrastructures and processes.

☆This work was partially supported by the German Federal Ministry of Education and Research (BMBF) within the collaborate research project DINGfest

*Corresponding author

Email addresses: michael.kunz@ur.de (Michael Kunz), alexander.puchta@ur.de (Alexander Puchta), sebastian.groll@ur.de (Sebastian Groll), ludwig.fuchs@nexis-secure.com (Ludwig Fuchs), guenther.pernul@ur.de (Günther Pernul)

Preprint submitted to Journal of Information Security and Applications November 13, 2018

While at the beginning only a small number of compliance standards and regulations had to be met (e.g. SOX [1], Basel II, Basel III [2]), nowadays, governments and organizations are more and more imposing compliance requirements that can only be governed by standardized IAM processes, guidelines, and technologies [3]. Furthermore, while initially only basic automation features were relevant, benefits related to process acceleration by facilitating efficient IAM measures play an increasing role for modern companies [3].

Today's IAM solutions already allow for a successful automation of most user administration workflows, offer dedicated functionality for security analyses, and at the same time deliver federation services and cloud-integration. They mostly employ role-based access control (RBAC) [4] as their underlying access control model. RBAC allows for a reduction of complexity by bundling permissions and employees into roles [5]. However, at the same time this can lead to steadily increasing role numbers and role administration efforts. In order to overcome these limitations, recent research as well as practical implementations are following the notion of attribute-based access control (ABAC) [6]. In contrast to RBAC, ABAC is more flexible and allows for the depiction of both, fine-granular and coarse-grained access rules [7]. ABAC checks the values of subjects', objects' or environmental attributes against pre-defined rules and allows or denies access based upon the fulfillment of these. Correctly maintained attributes (such as employees' business functions) do not only simplify entitlement provisioning but also support the discovery of violations of the least privilege principle. IAM workflows such as on-/off-boarding/movers of employees are easier covered by policies based on attributes rather than using static roles.

The success of any ABAC implementation, however, heavily relies on the underlying processes for a structured management of attribute definitions and attribute values. This task has not been receiving much attention within the research community up to now. Only few authors have pointed out that successful attribute management is a mandatory requirement for dynamic systems relying on attributes (e.g. [8]). Erroneously assigned attribute values can lead to unwanted access, effectively representing security risks and ultimately allowing intentional or unintentional abuse by insiders. Research offers several general data quality frameworks. However, these approaches do not offer the guidelines nor the fine-grained implementation details needed (e.g. quality metrics) for improving attribute quality in the context of IAM. They typically give high-level, non-IAM related guidance regarding the structure of quality improvement processes and do not provide specific recommendations or tool-support. At the same time, existing data quality metrics only provide generic support for attribute quality measurement and are not embedded into a process-oriented attribute quality concept. None of the existing approaches, for instance, offers a consistent overview of existing attributes within the various application systems connected to a centralized IAM infrastructure. They do not provide information about which attributes are incorporated into access policies, which access policies should be re-engineered, and which attribute values need to be investigated due to low data quality.

We argue, that due to the severe IT security risks imposed by low attribute quality, a structured and applicable approach to attribute quality management for IAM is needed. We thus introduce TAQM (**T**otal **A**ttribute **Q**uality **M**anagement), a dedicated attribute quality improvement approach tailored to the characteristics of IAM and attribute-ttribute-based access control in the remainder. In order to do so, we firstly introduce a generic, conceptual IAM model as the foundation of our research activities and to shape the scope of TAQM. Secondly, we analyze existing data quality management approaches regarding their suitability for IAM environments. Based on the results, we propose our novel approach TAQM for assessing, maintaining, and improving IAM attribute quality (Section 4).

The remainder of this paper is structured as follows. In Section 2, we outline our addressed problem and related work is provided. Section 3 describes our used research methodology as well as our contribution to the field. Afterwards, we present the IAM model (Section 4) which supports in discovering evaluation criteria for the analysis of existing data quality management frameworks in Section 5. Consequently, a comparison and selection of those frameworks is carried out in the same chapter. After this step, our attribute management approach is presented in Section 6 by integrating core elements from the chosen quality management frameworks with automation tools and procedures developed based on our IAM experience from industry projects. Within Section 7, the prototypical implementation of selected measures and optimization tools as part of the existing IAM analytics & cleansing platform *Nexis Controle*[1] together with a feasibility analysis within a real-world use case evaluation is provided. We conclude with known limitations of our approach and provide an outlook for further research in Section 8.

2. Problem and Related Work

Extensive research considering IAM processes, IAM policies and their implementation, as well as the underlying access control models has been carried out in the past [9]. RBAC, for instance, has evolved as the de facto standard for managing the access of thousands of employees to IT resources in many companies [10]. Following this concept, permissions are bundled into roles which are subsequently assigned to employees. This reduces administrative efforts but at the same time can lead to a steadily growing number of roles [11] while offering only a limited flexibility regarding contextual changes (e.g. departmental changes of employees) [12]. Furthermore, studies have shown that RBAC implementation costs an average of 2,410,000$ for a company of 10,000 employees [13]. As a result, ABAC has gained attention in both, research and practical application over the last years. ABAC leverages attribute definitions in order

[1] `https://www.nexis-secure.com`

to model dynamic access management policies based on attribute values of entities like employees or permissions. Initial ABAC approaches were introduced by Priebe et al. [14] and Yuan et al. [15]. A more comprehensive view on ABAC has been given by Hu et al. [6]. Research already provides approaches for initially developing or re-designing policies for IAM in a time-efficient and complexity-reducing manner (cf. [16, 3]). However, ABAC models heavily rely on the completeness and correctness of the underlying attribute values used within those policies. Consequently, a structured approach for maintaining attribute data quality is needed by organizations utilizing ABAC. Consider the following simple example[2] clarifying the addressed problem of attribute quality within ABAC environments:

Table 1 deals with different data quality problems relevant for IAM. It shows an excerpt of identities within an IAM system having the attribute "location" (working place of the employee) and "cost center" (used for internal accounting). Consider an additional ABAC policy granting access to the relevant file storage if and only if the employees' "location" equals *Munich*. One can identify two typical data quality problems within the table, as one "location" is shortened to *MUC* for employee #2 while #3 is completely missing a "cost center". According to the existing ABAC policy this heavily restricts the access to relevant resources for employee #2 as he does not fulfill the policy. Errors like these can for example arise if entries are inserted manually (e.g. by HR staff entering wrong identity information). Additionally such attribute data often does not get revised as it is seen as an unnecessary or too extensive task leading to a declining attribute quality. Thus ABAC can not be applied efficiently and identities are hindered in executing their work or circumvent such policies through direct assignment of permissions which can negatively affect security. In order to solve such problems this paper applies an attribute data quality management approach specifically tailored to existing IAM requirements.

Table 1: Example for IAM data quality problems

ID #	First Name	Last Name	Location	Cost Center
1	Yasmin	Olivid	Munich	Cost Center 1
2	Henry	Zellers	MUC	Cost Center 1
3	Charles	Ellsworth	Munich	

Over the last decades, a large body of work has been conducted in the field of quality management in general. Various notions of quality can be differentiated, e.g. *quality management* in general, *data quality management*, and *attribute quality management* [17, 18]. General quality management focuses on the quality of physical products while data quality management is specifically

[2]In the remainder of this article we mainly exemplify attribute quality for identities (e.g. employees) as this can be understood quite intuitively. However, all relevant elements within IAM may have respective attributes and are concerned with attribute quality (cf. Section 4)

dealing with managing the quality of structured data [19]. Additionally, within the research community the term information quality management is used for bundling activities related to the quality management of unstructured data [19]. However, as IAM in general handles structured data (e.g. employees and their master data, permission master data and assignments), information quality is of minor importance for our research.

Especially within the field of general quality management, extensive research has been published [20, 21]. Pioneering in the area of total quality management, Deming laid the foundation for modern quality management [17]. The defined principles are still incorporated into several quality management systems like the Baldrige Performance Excellence Program[3]. In addition, other approaches like lean management or six sigma were introduced [22, 23]. They added further dimensions to Deming's total quality management. However, none of these approaches provides a comprehensive integration of the notion of data quality [18]. As a result, dedicated data quality approaches try to cover these aspects by focusing on digital data quality. Batini et al. provide a comprehensive overview of existing data quality approaches [19] which provides the baseline for our overview of approaches in Section 5. Beside the existing data quality frameworks, researchers also have dealt with various data quality metrics over the last years. Using statistical or mathematical concepts, the quality of information and data sets can, for instance, be determined and compared [24]. There is a set of metrics defined for quantitative measurement of data quality. Examples would be metrics for consistency or timeliness [25, 26, 24] or techniques for detecting duplicate data entries [27]. They are also partly considered in the data quality frameworks mentioned above. Of course there exist many more different metrics yet we just wanted to give a hint on this topic.

However, attribute quality management has not been researched in the past to a sufficient extent. A recent identification of areas of research within ABAC does not even list attribute quality management as an individual category [28]. Other authors are aware that data quality is connected with attributes within IAM but do not provide any valid solutions [29, 30]. None of the provided approaches satisfies the requirements of IAM environments: While quality metrics rather aim at providing isolated mathematical means instead of dealing with structured data quality management processes, existing quality management or ABAC approaches fail to provide concepts for the evaluation, strategic management, and optimization of attribute definitions and attribute values. Consequently, organizations are missing a comprehensive approach for measuring, maintaining, as well as improving their IAM attribute quality.

In the following we aim at closing this research gap by defining a general attribute quality management approach for IAM environments. It does not only provide a generic high-level methodology but also integrates specific metrics for attribute quality handling. By doing so, it satisfies the demand for providing metrics suitable for attribute quality evaluation within ABAC while at the same

[3] `https://www.nist.gov/baldrige`

time offering an integrated process-oriented approach that can be applied to large-scale IAM scenarios.

3. Methodology

The underlying research methodology is displayed in Figure 1 and based on the principles of Hevner et al. [31]. Our facilitated knowledge base covers the foundations and methodologies from the two fields of IAM and data quality management. Business needs for increasing attribute quality within IAM systems are the baseline for our environment. On this basis we design a novel approach for evaluating, maintaining, and optimizing IAM attribute quality. We experimentally evaluate a prototype implementation of our solution consequently.

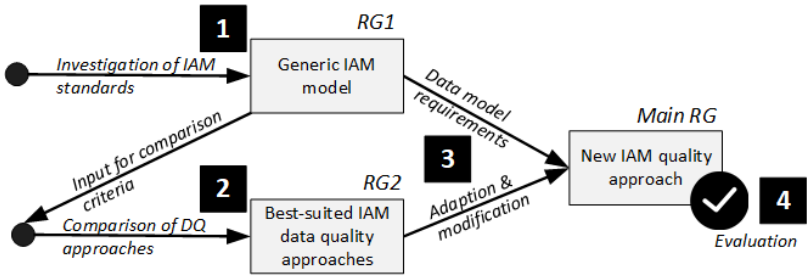


Figure 1: Applied research methodology throughout the paper

In order to achieve our research goals (RG), we firstly derive a generic conceptual IAM model (1) based on existing literature as well as project experience in order to get a comprehensive picture of IAM-relevant entities. The IAM model (RG 1) serves as input for establishing specific IAM quality requirements which can be used for the evaluation and comparison of existing data quality management approaches (2) regarding their suitability for IAM environments. The results (RG 2) then are inspected in detail and complemented with IAM-specific requirements, measures and optimization efforts (3). By combining elements of the approaches and specific requirements of our IAM model we generate a novel IAM attribute quality approach and a prototypical implementation (Main RG) which in turn is evaluated using real world data in order to demonstrate its feasibility and effectiveness (4).

4. Conceptual IAM Model

In the following, we compose a basic IAM model including all main entities relevant within the context of IAM. Note that in this paper, the term entity refers to any object whose attributes or master data are managed or used by an IAM system. The proposed model represents a minimal approach of mandatory

elements integrated in modern ABAC-based IAM environments and thus acts as the foundation of our novel IAM data quality management approach. Additionally, it represents the scope that needs to be addressed by the new approach. In order to come up with such a conceptual IAM model, we investigated the most relevant IAM standards and technologies: LDAP [32], SAML/Shibboleth [33], OAuth [34], SPML [35] and XACML [36] (for an overview cf. [37]). In order to reflect its practical relevance, SCIM [38], a relatively new industry standard already adopted by existing IAM products, was additionally included (even though it has not been listed in the aforementioned survey). For each standard, we extracted the covered entities relevant for IAM environments together with their mutual relationships and listed them in Table 2 within the third column. During a manual pre-selection process, standards which do not include any entities relevant for a comprehensive conceptual IAM model due to a different application focus (i.e. WS-Federation, CoSign, OZ, CAS, OIDC and Kerberos) have been excluded from further analyses. Kerberos, e.g., focuses on a client-server communication protocol rather than describing IAM entities. Furthermore some of these excluded standards are based on other standards (e.g. OIDC is based on OAuth) and would only include already identified IAM entities. In addition, several sub-models of standards like RBAC have been developed. They introduce additional concepts such as task-inclusion [39] or dedication to an organizational-structure [40] into the original RBAC model [9]. However, the goal of our conceptual model is to present the basic entities relevant for IAM in order to act as starting point for general quality management criteria - and not to act as generic model covering each IAM application scenario.

Based on all the input within Table 2 we crafted a more generalized model with the help of "derived IAM entities" and defined mandatory (digital identity, account, permission) and optional (role, policy, context, attribute) entities (cf. Figure 2 for the overall model). We made the transition from relevant entities to derived IAM Entities by listing all named entities. We then grouped the entities according to the following elements based on the respective standard description and our experience in IAM. For instance, LDAP speaks of persons and organizational persons. We therefore created the entity "(Digital) Identity" and matched it with the LDAP-specific terms and continued this for all standards:

(Digital) Identity. In organization-wide IAM scenarios, digital identities are the representation of human users, e.g. within the personnel management system (cf. [41, 42]). Some organizations operate multi-identity models in which one real-world employee is represented by several digital identities (sub-identities).

Account. Digital Identities are in turn represented within the target systems by user accounts (following the concepts from LDAP or SCIM). Note that several existing standards (e.g. SAML, SPML, XACML) do not differentiate identities and accounts because of their limited application-specific focus. Within IAM,

Acronym	Description	Relevant Entities	Derived IAM Entities
LDAP	Directory protocol for storage of user accounts and group memberships [32]	person, user, inetOrgPerson, organizationalPerson, group, groupOfNames, groupofuniquenames	Digital Identity, Account, Permission
SAML / Shibboleth	Exchange format for authentication and authorization information. Also attribute information can be exchanged. [33]	Subject, Attribute, AttributeQuery	Identity, Account, Attribute, Policy
SPML	Request and response protocol for provisioning information of accounts and resources [35]	Requestor, Provisioning Service Object, QueryClauseType	Identity, Permission, Role, Policy
OAuth	Standard for access delegation, typically for website access [34]	Resource	Role, Permission
SCIM	Standard for exchanging user and employee information within the cloud [38]	Identity, User, Singular Attribute, Multi-valued Attribute, Simple Attribute, Complex Attribute, Group	Digital Identity, Account, Attribute, Permission, Role
XACML	Standard for a policy language, that allows for a fine-granular and attribute-based expression of policies. Also offers an architecture and protocol for the interpretation of the language [36]	Attribute, Context, Policy, Policy set, Rule, Subject, Resource	Attribute, Context, Policy, Digital Identity, Account, Permission, Role
RBAC	Access control model that is currently de facto standard. It groups identities/accounts and resources/permissions together to roles [4]	User, Permission, Session, Role	Digital Identity, Account, Permission, Context, Role, Permission
ABAC	Access control model that founds authorization decisions on rules based on attributes [8]	Subject, Resource, Attribute, Environment Condition, Policy, Rule	Digital Identity, Account, Attribute, Permission, Role

Table 2: Table with investigated standards for the creation of a basic conceptual IAM model

an account represents the entity with which an employee logs on to a specific application system (e.g. LDAP Directory, SAP, etc.).

Permission. Accounts are typically assigned to permissions in order to grant access to certain IT resources. A permission is considered as any access right within a specific application system, irrespective of its granularity. It consists of an authorization statement and an object this authorization is granted for (e.g. following the original RBAC model [4]) and can be hierarchically aligned. For instance, a file share "Marketing Campaign 2018" can be typically accessed via read, modify or update rights. In this case, "Marketing Campaign 2018 - read" is an example for a permission. This permission can optionally be nested into another more generic permission "All Marketing Campaigns - read".

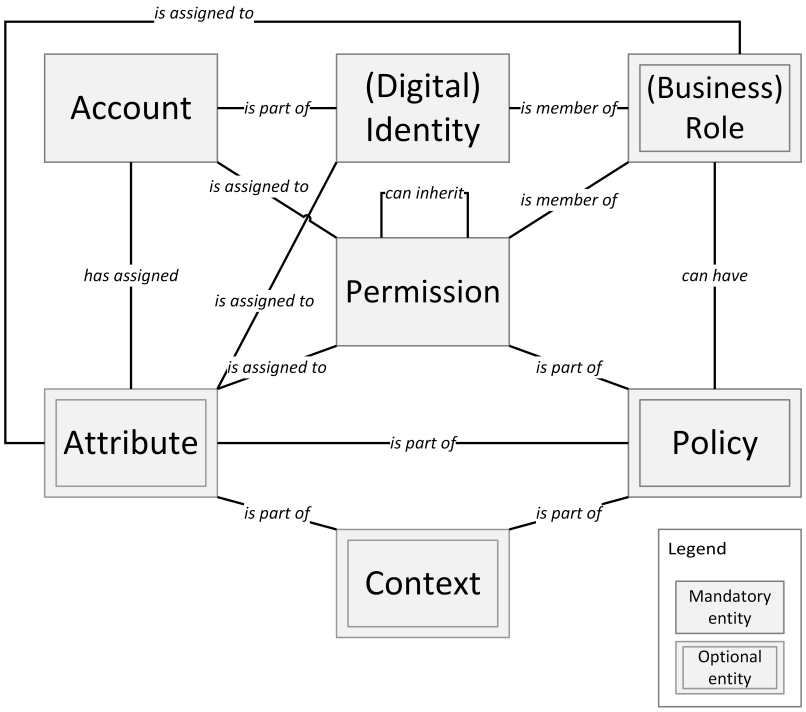


Figure 2: Conceptual IAM model based on IAM standards

Business Role. RBAC as the currently predominant access control model defines the concept of roles, which are interpreted as business roles in the IAM environment. Business Roles are used to bundle permissions from target systems (e.g. SAP roles, Microsoft Active Directory groups, or mainframe profiles) into

a business-relevant object (e.g. business function of an employee). In contrast to permissions, they are typically instantiated within IAM systems in order to allow an system-independent view on a meta-level.

Policy. A policy represents a rule or a set of rules that defines whether a role or permission is assigned to an account or identity. This follows the definitions known from ABAC, SPML and XACML.

Context. Context represents the relevant scenario a policy can be part of. In terms of ABAC, for instance, environmental conditions such as the time zone or geologic location during a log-in represent a certain context.

Attribute. Attributes represent the meta-data related to accounts, identities, roles, policies, and permissions and are in turn assigned to a specific context. SCIM, XACML, and ABAC are, for instance, heavily depending on attributes. Table 3 shows typical examples of attributes for the entities defined in our model. Note that companies usually define additional custom attributes to fit their needs.

Table 3: Typical examples for Attributes in the IAM Context

Entity Type	Attribute Example
(Digital) Identity	Department, Job Title, Location
Account	Account Type (e.g. Admin, User), Target System
Permission	Criticality, Target System
(Business) Role	Criticality, Business Function

Contribution to the attribute quality approach. With the help of the IAM data model we clarify the entities and the respective attributes which have to be considered for an attribute quality management approach. Using this model one can easily see which entities of IAM have to be included into a comprehensive attribute quality management approach and thus defines its scope. Otherwise important aspects (e.g. context information) do not get included in such approaches when it comes to practical implementation. The model itself also serves as an important requirement for the approach and is used as evaluation criterion for the existing data quality management approaches in Section 5. Finally we ensure that the underlying data model of our approach is correct and includes all relevant objects used in relevant IAM standards.

5. Selection & Evaluation of Data Quality Management Approaches

In the following we investigate existing data quality management approaches in respect to their suitability to serve as foundation for structured IAM quality management. The evaluation is then executed on the basis of criteria derived from both, our IAM data model from Section 4 and general IAM conditions.

5.1. Selection of Data Quality Management Approaches

During a first step, we collocate a list of potentially suitable approaches known from literature. Batini and Scannapieco provide a comprehensive overview of traditional approaches for data quality management [19]. Note, that two approaches mentioned in their publication were excluded of our analysis: CIHI is an approach [43] purely focusing on administrative databases within the Canadian health care sector. Similarly, ISTAT [44] is solely used for Italian public administrations and the improvement of address data of citizens and businesses. Due to their specific application scenario, both approaches are not suitable for our purpose. Furthermore, we extended the list of [19] with QIAM, to the best of our knowledge, the only approach specifically focusing on data quality within IAM so far [45]. The full list of investigated data quality management approaches can be found in table 4.

Table 4: Overview of data quality management approaches

Data Quality Management Approach	Reference	Focus
AIMQ	[46]	Data Quality Assessment
AMEQ	[47]	Data Quality for Mechanical Products
CDQ	[48]	Generic Data Quality Framework
COLDQ	[49]	Costs of Low Data Quality
DaQuinCIS	[50]	Data Quality for Cooperative Information Systems
DQA	[51]	Data Quality Assessment
DWQ	[52]	Data Quality for Data Warehouses
IQM	[53]	Data Quality for Web Data
QAFD	[54]	Data Quality for Financial Data
QIAM	[45]	Data Quality for IAM
TDQM	[18]	Generic Data Quality Framework
TIQM	[55]	Generic Data Quality Framework

5.2. Description of Data Quality Management Approaches

In the following the aforementioned approaches are briefly introduced. *AIMQ* introduces a questionnaire to collect and analyze data. Based on these results, activities for improvement are identified. Within *AMEQ* an activity-based approach is used to measure data quality for mechanical products. *CDQ* consists of a comprehensive data quality framework based on business processes. Within this approach, the effects of low data quality on business processes are analyzed. *COLDQ* investigates the cost of low data quality for organizations by means of a scorecard based approach. *DaQuinCIS* focuses on cooperative information systems with an e-government context. Their D^2Q model is used to define data sets and quality properties. The *DQA* framework focuses on the identification

and comparison of subjective perceptions of individuals and objective measurements of data quality. Based on this, differences regarding the quality can be deduced resulting in tasks for improvement. The *DWQ* approach is one of the first methods related to data quality management. It is centered around data quality for data warehouses and leverages queries to determine the quality of a data warehouse. On the contrary *IQM* concentrates on web-specific data quality. This approach combines different existing tools for websites to determine the data quality. *QAFD* is used as a framework for data quality regarding financial data. It introduces initial quality measures for financial application scenarios. *QIAM* is the first approach to introduce structured data quality management in IAM. However, it is specifically focused on data quality for role-based access control systems, essentially limiting its view and applicability in ABAC environments. The *TDQM* approach is the first comprehensive data quality framework to be introduced. It is based on a process-driven cycle analogously to the Total Quality Management by [17]. Another generic data quality framework is represented by *TIQM*. In addition to commonly used processes within data quality management, *TIQM* establishes a specific process for culture transformation.

5.3. Definition of Comparison Criteria

Within this and the following sections the previously introduced quality management approaches are compared regarding their applicability in IAM environments (see Table 6).

Our evaluation is based on criteria specifically relevant within attribute-based IAM environments. These criteria can be understood as IAM-specific requirements for attribute quality and have been derived from both, research publications and experience from various real-life IAM projects as well as from our IAM data model (cf. Section 4). Note that our goal was not to define an exhaustive list of criteria, but rather to come up with a selection of requirements of major importance for IAM. We argue that only quality management approaches that cover those basics are suitable for application. Table 5 lists the selected comparison criteria, followed by a short description and discussion.

Focus on Attribute Quality. Approaches suitable for IAM environments need to be centered around data quality and its improvement. Several existing approaches only deal with the flow of information and do not explicitly focus on the quality of (data) attributes (e.g. [49]). This, however, is a core requirement within data-driven IAM environments in which attributes are critical for the interpretation of attribute-based access policies (e.g. correctness of employee attributes or entitlement attributes).

Governance. Governance is of high importance in today's IAM infrastructures for organizing and structuring an IAM's performance. Therefore, a data quality approach for IAM needs to be capable of integrating organizational responsibilities and tasks during quality measurement and improvement. For instance, actions regarding quality management need to be audited in a secure manner (e.g. who decided which employee was assigned to a certain attribute value).

Table 5: Used criteria

Criteria	Description
Focus on Attribute Quality	Data-centered perspective on attribute quality instead of e.g. concentration on data flows
Governance	Integration of long-term management processes such as responsibilities for entities and related tasks
Iterative Approach	Ongoing process-oriented approach with repetitive phases for incremental improvement of data quality
Granularity	Level of detail of quality measures of the respective approaches
IAM Content	Inclusion of IAM relevant topics and requirements within the framework
IAM Completeness	Capability to integrate all entities known from the conceptual IAM model

Thus the notion of responsibilities by IT- as well as non-IT staff for actions and tasks needs to be considered.

Iterative Approach. Data quality management in IAM environments needs to be based on a strategic and iterative methodology as entity-related and organizational structures constantly change during the development of an organization. It has to be capable of integrating new data elements (e.g. new IT systems getting on-boarded) throughout an evolving attribute quality management cycle. As a result, every suitable approach has to provide a structured and iterative process for data quality (e.g. a PDCA (Plan-Do-Check-Act) cycle).

Granularity. Granularity describes the level of detail a certain quality management approach incorporates. While many deliver a generic process model (e.g. [49, 53, 51]), specific details of actual technical measures are often missing. For ABAC, implementation guidelines with a deep level of technical context already exist [56, 57]. We argue that for successful adoption, combining both worlds is required. While high-level processes can guide strategic development, technical implementation details offer guidance and ease real-life implementation.

IAM Content. Within this criterion, we assess the extent to which an existing approach already covers IAM-specific topics. There is already a wide range of different requirements relevant for IAM (e.g. Identity Life Cycle Management) that can be leverage for this criterion [58, 59]. In case an existing approach covers or addresses certain IAM functionalities out of the box, it might be adopted easily to form a basis for structured and generic quality management in IAM environments. Please note that this criterion is aggregating a range

of different topics to avoid defining a single criterion for each specific IAM functionality.

IAM Completeness. Existing approaches typically cover general data quality requirements or data quality dimensions [18, 46]. In respect to IAM, they need to be able to cover all entities and relationships included within the previously presented conceptual IAM model. The entities should either be explicitly mentioned or the approach should at least be flexible enough to consider all entities. The same holds for all relationships between the entities (e.g. the binding of attributes to permissions). As the introduced conceptual IAM model represents a minimal approach of mandatory elements, we argue an existing approach needs to be capable to deal with those entities in order to be suitable for structured long-term quality management.

5.4. Evaluation Summary

In order to evaluate the presented attribute quality management approaches we applied a Likert Scale from 0 to 4 points (equally from ○ to ●) [60]. For each criterion we rated the approaches relatively to each other and selected the top-rated three approaches (i.e. the top quartile) for further usage during our research. The final analysis is displayed in Table 6 with the top quartile approaches marked in green. They are selected to act as baseline for our new quality management approach presented in the following section.

Table 6: Comparison of data quality management approaches

Approach	Focus on Attribute Quality	Governance	Iterative Approach	Granularity	IAM Content	IAM Completeness
AIMQ	◑	◔	○	◔	◔	◑
AMEQ	◕	○	◕	◕	○	○
CDQ	◕	◔	◕	◑	◑	◑
COLDQ	◔	◑	◑	○	○	○
DaQuinCIS	●	○	●	◔	◔	◔
DQA	●	◔	○	◔	◔	◑
DWQ	◕	○	○	◑	○	◔
IQM	◑	○	●	○	○	○
QAFD	◑	◔	◕	◑	○	○
QIAM	◔	◕	●	◔	●	◕
TDQM	●	◕	●	◔	◑	◕
TIQM	◕	◕	●	◑	◑	◕

While generic requirements like *Focus on Attribute Quality* and *Iterative Approach* are rated quite well on average, IAM-specific requirements (e.g. *Governance* or *IAM Content*) are not addressed sufficiently.

The results reveal three existing approaches with a similar performance, dominating the other quality management approaches: TDQM, TIQM (each 17 points), and QIAM (16 points). Therefore we did not decide to take only one approach and improve it with attribute quality techniques as we did not want to resign the benefits of the other two well rated approaches. However, none of those approaches cover all IAM-related requirements to a sufficient extent. QIAM representing an IAM related approach for example lacks the focus on attribute quality as it was mainly developed for a role based environment (e.g. for SAP roles). Furthermore it does not offer an existing toolset of metrics like other approaches do. Regarding TDQM and TIQM it is exactly vice versa as they lack sufficient IAM content and granularity.

6. The TAQM Approach

The comparison and evaluation of existing frameworks above has shown the inability to cover core requirements needed for structured attribute quality management in IAM (cf. Section 5). To overcome this gap we present TAQM (**T**otal **A**ttribute **Q**uality **M**anagement), a novel attribute quality management approach for IAM environments (see Figure 3). Its main characteristics are:

- Cyclic execution in order to cope with the dynamic nature of IAM data
- (Semi-)automated tool support (e.g. during discovery of attribute errors)
- Integration of human experts to foster existing organizational knowledge
- Data-centricity that focuses on correctness of data in order to support the improvement of quality management and policies
- High-level structure and low-level guidelines for a fast and easy deployment

TAQM is aiming at supporting both, the technical and organizational nature of IAM. It follows the core concepts for IAM data quality presented by Fuchs and Pernul (QIAM, [45]) as well as those of the TDQM and TIQM frameworks by Wang et al. [18] and English et al. [55]. Following Wang et al.'s well-respected method of defining, measuring, analyzing, and improving quality, TAQM incorporates four cyclic phases (see Table 7). Additionally, it employs the process orientation known from TIQM in order to structure each execution phase and introduce automation support. Furthermore, we complement TAQM with IAM-specific activities and supportive automation tools in order to increase adaptability. Each of its four main phases thus provides low-level implementation guidance in order to overcome the limitations of already existing purely high-level approaches.

Note that the automation techniques presented are not exhaustive and might be extended in future work. However, we argue that they already cover the basic required tasks and thus present a valid baseline for TAQM implementation. The evaluation in Section 7 underlines this assumption in real-life scenarios.

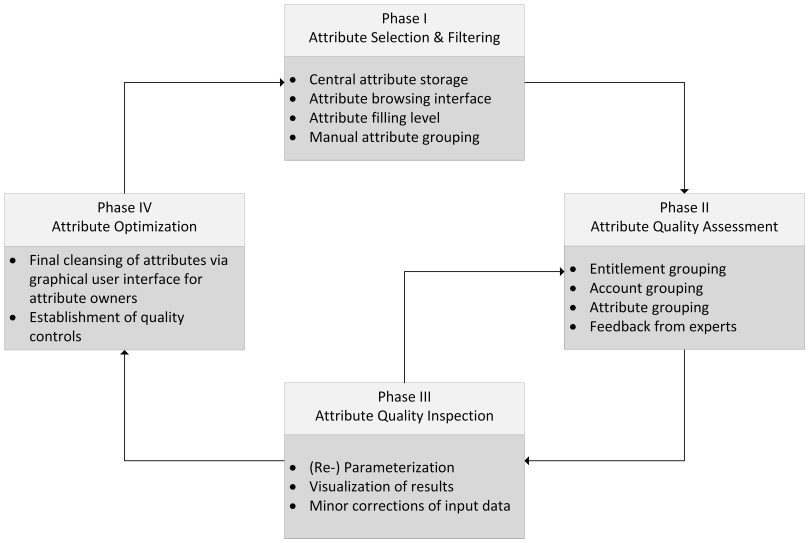


Figure 3: Proposed TAQM approach

Table 7: Phases of TAQM

TAQM Phase	Corresponding Phase in [18]
Attribute Selection & Filtering	Define Phase
Attribute Quality Assessment	Measure Phase
Attribute Quality Inspection	Analyze Phase
Attribute Optimization	Improve Phase

6.1. Attribute Selection & Filtering

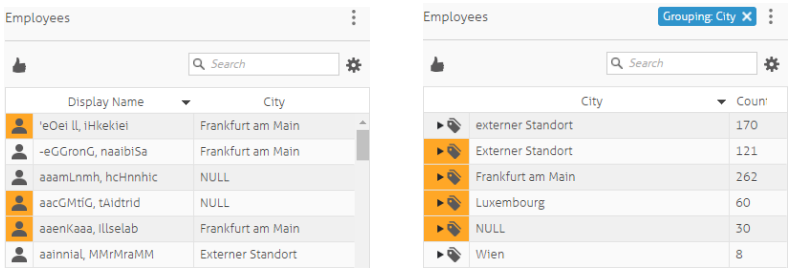
During this initial phase, human experts select the set of attributes and IAM entities relevant for a specific company (cf. Figure 4). For instance, attribute-based access policies often rely on employees' attributes such as the employee type or work location while permissions usually carry ownership or information like their risk-rating. In order to allow for an efficient selection process, TAQM suggests the following automation tools during Phase I:

- Central attribute storage based on the proposed conceptual IAM model(cf. Figure 2)
- Attribute browsing and selection interface
- Attribute filling level analysis
- Manual attribute grouping functionality

Figure 4: Simplified overview of the *Attribute Selection & Filtering* phase

First of all, relevant attributes need to be loaded into a centralized attribute storage (cf. Section 4). This storage is responsible for handling the connection between entities and their attributes and acts as basis for data analysis measures. Note that existing IAM systems already offer a centralized database for managing the execution of operational IAM processes like the onboarding or offboarding of employees. They, however, lack the functionality of structured attribute management including data browsing, analysis, and dedicated attribute-related processes. We thus argue existing IAM implementations need to be extended with a dedicated centralized attribute storage for this purpose. To overcome this problem the attribute storage can be used as the master system regarding all attributes while IAM systems just need to manage the attribute values accordingly (e.g. the attribute storage defines a certain range of valid values. This range could be queried by the IAM system and the values can be saved respectively). Additionally traditional IAM databases can not be changed so easily compared to a dedicated and separated attribute storage. We recommend to design this storage based on our proposed conceptual IAM model (cf. Figure 2). The model provides a consistent foundation for executing the steps of TAQM and contains the required entities and relations. From a technical point of view it ensures that the proposed visualizations and algorithms can be implemented and executed. Furthermore people applying TAQM should be familiar with the model, as it is based on various IAM standards. Therefore visualizations (grouping, aggregation, etc.) based on the model are easy to understand for human experts. This is especially important as experts have to make various decisions based on such visualizations while executing the steps of TAQM.

Secondly, data browsing interfaces are required to improve the attribute analysis and selection process by human experts. In real-life implementations with a high number of attribute definitions and values combined with several thousand instances (e.g. employees or permissions) such support is mandatory. The data browser should further be enriched with automated attribute filling level analyses (following the metric published by [61]) in order to detect major syntactic attribute quality issues like NULL-values. In many scenarios, empty attribute values are unwanted and hint at potential process flaws which prevent access control policies from correct interpretation. For extended discussions on such NULL-values and their handling, see Heinrich et. al [61].



(a) Employee attribute value list (b) Grouped employee attribute value list

Figure 5: Attribute NULL-value analysis (anonymized employee names)

Thirdly, data grouping functionality is required to support human experts when analyzing the distribution of attribute values and gain insight about possible source data quality issues (see [62]). It reveals syntactic data errors (i.e. typos, inconsistencies etc.) and allows for a semantic analysis of attribute value distributions. Our experience in practical projects revealed that many organizations are not aware of current attribute definitions and used attribute values. This typically stems from a long history of decentralized attribute management processes carried out for systems individually. Figure 5 visualizes a single-view (a) and grouped-view (b) example of employees' work location which could be integrated in a supportive tool for TAQM execution.

Finally, the centralized attribute storage should offer a graphical attribute selection interface. According to Wang et al. this supports human experts choosing the desired set of attributes relevant for further improvement [18]. During the first execution cycle of TAQM, we recommend to only select a basic set of the most important attributes. These attributes can subsequently be evaluated by applying different constraints like a maximum number of characters for a string attribute. As IAM is heavily relying on organizational anchoring [41], we argue that an ownership concept for IAM entities should be introduced within this phase (similarly to role owners as shown in [10]). An attribute owner acts as the primary contact person for an attribute and its values and is responsible for its maintenance. A similar definition can be also found in [18].

6.2. Attribute Quality Assessment

After attribute selection, an initial semi-automated assessment of the current attribute quality needs to take place in order to foster later human analysis during Phase III (depicted in Figure 6). The output are possible anomalies included within the analysis results. Note that TAQM also allows for a subjective manual quality estimation by attribute owners or responsible staff. Using this method known from [51] allows us to identify deviations from subjective and objective perceptions of attribute quality. However, in general, assessment automation typically is required in IAM scenarios with a very large number of

attribute values and assignments. As general data quality metrics already were

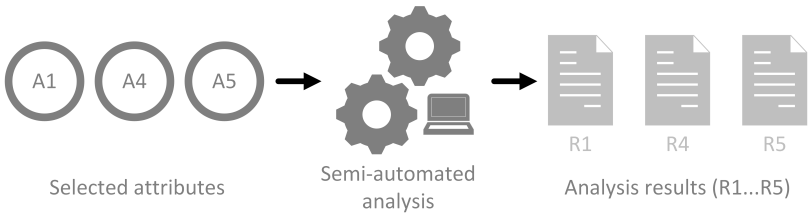


Figure 6: Simplified overview of the *Attribute Quality Assessment* phase

subject to profound research in the past [18, 46, 51], we are focusing on IAM-specific attribute assessment procedures in the following. In contrast to TDQM [18], for instance, we work with a predefined but yet extensible set of quality metrics specifically suited for IAM data. We rely on core concepts known from the field of role development [63], where clustering of employees according to access data is executed in order to identify suitable role candidates. However, in contrast to the limited role-oriented view in [63], we allow for the automated analysis of any conceptual IAM model entity. Attribute Quality Assessment is executed throughout three steps:

- Data grouping
- Data validity check
- Assessment execution

At first relevant data is grouped and correlation matrices are generated. These matrices are in turn validated for their applicability before a data quality assessment can take place.

Data Grouping. Firstly, all relevant entities are statistically grouped based on attribute values using two-dimensional matrices. The underlying assumption is that certain entity attribute values typically are well-managed due to their company relevance (e.g. "cost center" assignments of employees). Starting from such high-quality attributes, the highlighting of quality issues for other attributes can take place. Table 8 shows a practical example in which the number within each cell represents the number of employees having the same attribute values for both attributes. Imagine a "location" attribute of employees which has not been managed in a structured manner and data errors are expected. Following the example, employees might be grouped according to their "cost center" attribute which in turn is related to a second attribute dimension (e.g. the "location" attribute). Note that also missing attribute values can be handled, by collecting those NULL-values in a special group (see Figure 5). During the later assessment, outlier detection mechanisms can then highlight potential

Table 8: Simplified example for data grouping & validity check

		Attribute **Cost Center**	
		Cost Center 1	Cost Center 2
Attribute **Location**	Location A	98	100
	Location B	2	100
	Location C	0	50

data errors for the location attribute. Accordingly, other entities like entitlements, roles, and accounts can be grouped in order to detect quality issues using classification techniques known from [63].

Data Validity Check. After two-dimensional matrices for all relevant entities and their attributes have been created, their suitability for further analysis has to be confirmed. Two main issues limit the meaningfulness of a given matrix: i) Similar distribution of attribute values and ii) too few group memberships. In case of small groups (e.g. only two employees are assigned to a certain location; see location B in Table 8 row 2), further outlier analysis does not make sense. The same holds in case the attribute value distribution is not meaningful for a certain row or column in the matrix. Imagine the "Cost Center 1" consisting of 100 employees which all are assigned to different locations to the same extent. As a result, outlier detection is not able to determine a predominant location for this Cost Center. In order to automate the validity check, we apply threshold-based ratings related to group size and group distribution of each matrix row and column in order to define whether it is used during the subsequent assessment execution. More precisely, we validate if 50% of the most frequent attribute values cover 80% of the entities. Note that those thresholds can be configured according to the given scenario.

Assessment Execution. In the final assessment phase, TAQM identifies outliers and potential data quality issues. One automation technique, for instance, highlights all entities with attribute values in groups, where the overall distribution of the value is below a certain threshold (e.g. 5%). Following our previous example, Cost Center 1 consists of 100 employees out of which 98 are assigned to the location "A" while only two employees are assigned to "B" and none to "C". Those two outliers could hint at suspicious attribute values (e.g. an employee has a wrong location attribute). Contrarily, "Cost Center 2" would not indicate any anomalies. Similarly, imagine a number of financial-related permissions within a SAP system. A matrix calculated based on classification techniques reveals the various departments those permissions are used in. Based on that, it might be revealed that employees in the "Cost Center 2" department are wrongly assigned to those finance permissions.

6.3. Attribute Quality Inspection

After the assessment execution, the identified analysis results (containing possible quality issues) need to be reviewed by human experts in order to decide

if they indeed represent attribute quality errors or false positive alerts (see Figure 7). This can be cumbersome as a potentially high number of attribute analyses (each built on a different similarity matrix) needs to be conducted.

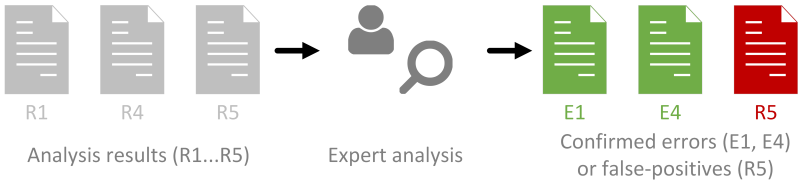


Figure 7: Simplified overview of the *Attribute Quality Inspection* phase

Typically, organizations attach a variety of different attributes to each managed entity within their IAM system. A large number of outliers detected within the numerous matrices might be the result. We hence argue that tool support is required for verification of outliers and suggest two techniques in order to achieve this:

- Visualizing coloured outlier matrices:
  Each of the matrices that resulted in suspicious attribute values can be visualized in a human-understandable form using outlier colouring. For example, orange or red highlighting (depending on the confidentiality level for a certain outlier), might hint at high-likely errors within the source data while green colouring shows standard attribute values. Group sizes and confidentiality ranges can further support human interpretation.

- An interactive grid visualization:
  Grid-based visualization techniques are able to display employee permission assignments within a two-dimensional matrix (cf. Figure 11). They have mainly been used for role development so far [64]. However, by highlighting specific attribute values of any entity, they allow for a contextual result analysis by offering data grouping, data coloring, or data filtering. For instance, grouping algorithms might discover results for a departmental head that has a single attribute value no one else is assigned to in his department. While this attribute value assignment might be conspicuous in terms of outlier detection, an interactive grid visualization can easily allow a human expert to identify this identity as departmental head (e.g. by displaying the department attribute of every identity).

Note that in comparison to TDQM [18], we allow a loop-back to Phase II at this stage. This enables an expert to quickly re-adjust data quality assessment mechanisms in case the outlier detection mechanisms have not been configured appropriately (e.g. include less attribute dimensions, change threshold values, or ask attribute owners to provide further semantic information about attribute values).

6.4. Attribute Optimization

After the identification of potential attribute quality problems and a first result validation by a human expert, the last phase of TAQM serves three main goals: On the one hand, it aims at cleansing identified errors (cf. Figure 8). On the other hand, it suggests the introduction of data quality standards as well as the set-up of strategic measures to maintain these quality standards.

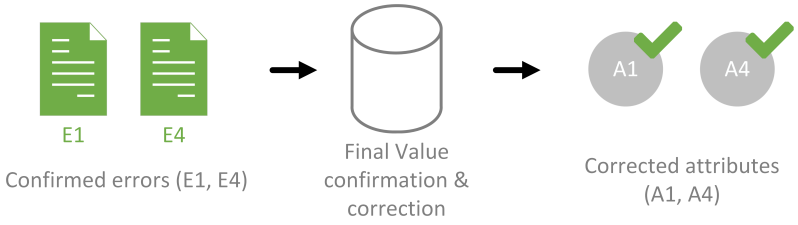


Figure 8: Simplified overview of the *Attribute Optimization* phase

Data Cleansing. Regarding data cleansing, human experts need to rate the outliers and propose correct attribute values or mappings for identified data errors. This could cover correcting current attribute values, assignments of user accounts to identities, or the clean-up of excessive permission assignments. Imagine a company where applications' user accounts have not been mapped to existing identities, i.e. employee master data from the HR system. During a first data cleansing cycle, automated analyses can identify which user account belongs to which employee based on attribute correlation regarding the different accounts, their assigned permissions, and the employee master data. After mappings have been automatically proposed (a user account with finance permissions in an SAP system could, for instance, be mapped to an employee within the finance department) and a human expert reviewed the results, a second execution cycle of TAQM might lead to further outliers which could not have been detected without this initial identity mapping (e.g. the attribute data of the user account is erroneous or the assigned permissions contain risks violating the principle of the least privilege).

Data Quality Standards. Regarding the establishment of and adherence to data quality standards, the knowledge of human experts can be fostered in order to gather semantic knowledge about the data. They can identify whether potential errors have technical or organizational reasons and make recommendations for data quality standards. Manually and decentralized attribute management, for instance, is very likely subject to a higher error rate than automatically derived attributes by a centralized department. As a result, not only the one-time clean-up of data errors, but a change in organizational or technical processes is required. In case organizational errors are the reason for user account attribute

issues, for instance, an IAM manager might define improved attribute management processes which include the definition of minimum data quality standards such as to prohibit NULL-values for IAM-relevant attributes.

Data Quality Maintenance. Besides such organizational change, technical measures can support the strategic maintenance of data attribute quality. Structured data reviews by attribute owners, departmental managers, or permission and role owners might be introduced. In the field of IAM, this typically is referred to as data re-certification and covers the human inspection of attribute values and assignments between entities from the conceptual IAM model. However, full data re-certification of attribute values as well as authorization assignments typically results in a significant organizational effort and thus costs. In order to minimize both, we propose a risk-based review of entities. For attributes with a large number of detected errors and a high impact on access policies, on the one hand, a full review might be reasonable. On the other hand, uncritical attributes or assignments might only partially be reviewed. Only suspicious data values might require periodic human evaluation in this case.

After completion of Phase IV the cycle restarts in Phase I. The now cleansed attributes together with previously gathered re-certification decisions can serve as an input for further optimizing other attributes or re-evaluating the quality improvement. Note, that TAQM by design allows for an on-demand parallel execution. Consider an organization that already cleansed a number of personnel master data attributes but is now forced by regulations to connect all locally-managed IT applications to their IAM system. Before this can take place, a comprehensive cleansing should be executed for all attributes that later might be included in attribute-based access policies. At the same time, another TAQM cycle might re-evaluate the previously cleansed personnel master data attributes in parallel. Even more, both execution cycles might involve different experts or be managed by different staff within the organization.

6.5. Fulfillment of Evaluation Criteria by TAQM Approach

After we outlined the main features of TAQM, the following section discusses them with respect to the previously introduced requirements (cf. Section 5 for data quality management approaches for IAM environments).

Focus on Attribute Quality. We purely focus on attribute quality (i.e. data quality of IAM attributes) as this is the foundation for IAM based on ABAC. However, one can argue that the TAQM approach could benefit from an integration of information flow components. We see this as a possible extension whereas we want to answer the initial question of how attributes for an IAM using ABAC can be improved.

Governance. Governance is a key element within IAM but it is not really integrated into data quality management approaches so far. Thus we strongly tie tasks and responsibilities within our approach to the organization and its individuals. The concept of defining attribute owners and to delegate tasks based

on structured re-certification processes (cf. Phase IV of our approach), supports the requirement of governed processes.

Iterative Approach:. TAQM is a cyclic approach, consisting of four different phases which are applied subsequently (including a possible quality assessment loop). By design, it fosters the iterative attribute quality refinement and thereby reflects the fact of IAM being an ongoing process with changing environmental conditions. Carve-ins or carve-outs of companies, movers (i.e. employees changing their position within the organization), or newly introduced IT systems result in a highly dynamic nature of attributes.

Granularity. Most of the existing data quality management approaches fail to deliver fine-grained guidance of how to improve data quality. TAQM does not only consist of a generic high-level phase model but also offers low-level automation metrics and data analysis tools for improving attribute quality within IAM. The previously introduced metrics and analysis techniques present a basic tool set that can be further complemented with individualized implementations.

IAM Content. TAQM is, to the best of our knowledge, the first comprehensive approach to manage attribute quality within an IAM environment and to deal with its specific characteristics. In order to achieve this, we combined existing data quality management approaches with IAM-specific requirements and included all relevant entities based on a structured data model. Additionally, TAQM focuses on the functionalities of IAM that are affected by poor attribute quality such as access regulating attributes.

IAM Completeness. The initially proposed conceptual IAM model serves as the baseline for the TAQM. While TAQM aims at increasing the IAM attribute quality in general, all available IAM entities are fully integrated. Attributes for all relevant and selected entities can be investigated and improved.

7. Applying TAQM

In the following, we evaluate our novel approach according to the design science research evaluation framework of [65]. We build our efforts on a naturalistic ex-post evaluation for rating the effectiveness of our socio-technical artifact using organizational access from our IAM project experience. We describe the application of TAQM throughout three real-life use cases from different companies. The required input datasets have been extracted from the companies' IAM systems and contain employee master data, organizational structure, user accounts, and entitlements from IT applications together with various attributes for all entities (see Table 9). The data was imported into a tool prototype automating the correlation analyses, data review process, as well as data cleansing during TAQM Phase II, III, and IV. Note that the central attribute storage of the prototype as well as most visualizations (e.g. Figure 11) for the experts are based on the proposed conceptual IAM model (cf. Figure 2). As stated in

Dataset	Pseudonym	Employees	Accounts	Entitlements	Applications	Attr. supplied	Date
#1	FactComp	19,829	17,308	7,018	1	8	08/2017
#2	FinComp	5,865	64,429	214,586	387	8	08/2017
#3	AutoCorp	11,386	17,698	75,274	1	15	09/2017

Table 9: Investigated IAM data sets

Section 6 this facilitates the technical implementation and improves the comprehensibility for the human experts. We facilitated the data analysis platform *Nexis Controle* provided by Nexis GmbH[4], a German IAM company and integrated our TAQM prototype functionality. Extending an existing software allowed us to facilitate available data import as well as workflow functionality (e.g. used during the review of identified outliers).

Note that the use cases for company 1 and 2 focus on the application of the first three TAQM phases (Attribute Selection, Quality Assessment, and Quality Inspection) as both companies executed a first TAQM cycle covering those phases in the year 2017. They are planning to execute phase IV subsequently. Use case 3 demonstrates the applicability of TAQM within an industrial company which is currently in the process entering Phase IV of our approach.

7.1. FactComp

The first use case covers a globally-operating manufacturing company with more than 12.000 internal and 4.000 external employees managed using a centralized IAM system which is connected to the main IT applications (Active Directory, SAP ERP, SAP HCM, amongst others). The company is currently improving security and user management efficiency by modeling attribute-based access rules in order to automate joiner, mover, and leaver processes. TAQM was employed to execute an initial attribute quality assessment for employee attributes which later are included in access management policies (e.g. every employee in the IT department is assigned to certain privileges automatically based on the "Department" attribute value).

Phase I. : At first, we imported the available HR master data together with user account and permission data stemming from the company-wide SAP ERP system (see dataset # 1 in Table 9). In total we received eight attributes out of which four were attached to the employee entity (employee group, managerial responsibility, employee type, and IT domain) and four to the permission objects. We were asked to only include permission data stemming from the

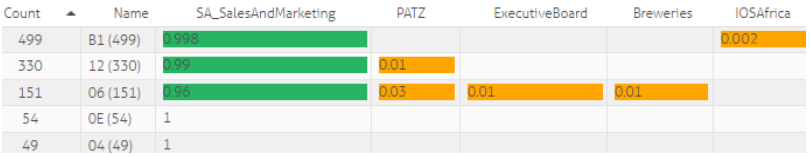
[4] `https://www.nexis-secure.com`

company-wide SAP ERP system for permission analyses as the company just recently completed a permission clean-up within this system and thus was able to deliver high-quality permission data. Due to space restrictions in this paper we only focus on the "IT domain" attribute which expresses the company area an employee is assigned to. An employee's IT domain is one of the characteristic attributes deciding about required access privileges. However, it is currently still manually maintained and thus error prone. In summary, 59 distinct attribute values assigned to each of the 19,829 checked identities were provided. Manual data review by IT experts confirmed that all attribute values were syntactically correct (e.g. no typos or spelling mistakes were detected).

Phase II. : During Phase II, we firstly analyzed all other employee attributes regarding their suitability for a correlation analysis regarding the "IT domain" attribute following our proposed approach from Subsection 6.2. The attribute "Employee Group" (revealing the management level of an employee), for instance, is already managed in a semi-automated manner by HR personnel due to its importance for payroll processes and thus suitable. We then automatically created all related classification matrices and attribute value groups for correlating the "IT domain" attribute of employees with their "Employee Group". This assessment led to a set of 184 conspicuous attribute value groups. Besides correlating employees' master data attributes, we also correlated the "IT domain" attribute with SAP ERP permissions assignments of the employees. This way, we were able to highlight employees which, according to their SAP ERP permissions, are likely member of a different attribute value group (e.g. an employee with access rights typical for the "IT domain" *Marketing* who is assigned to the "IT domain" *Sales*). This step resulted in a total of 58 possibly erroneously assigned value groups.

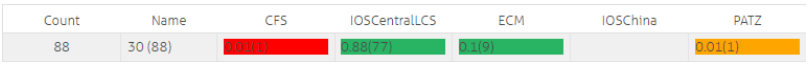
Phase III. : Together with IT experts, we reviewed the identified outliers for six "IT domain" areas. The example matrix for the Sales area is displayed in Figure 9, highlighting outliers using orange and red coloring depending on the level of significance (in this case 5%). The first column ("Count") lists the total count of members of a value group while the second column ("Name") shows the value of the respective attribute for the "Employee Group" attribute. The results show that within the Sales area there is only a small number of attribute values for the IT domain shown in the remaining columns (e.g. *PATZ*, *ExecutiveBoard*, ...). Note that the values stated represent the percentage of all value group members assigned to a certain IT domain. IT experts, for instance, confirmed during data cleansing that the attribute value *Africa* should typically be appearing only within the domain *International Operations*.

Another example presented in Figure 10 shows the distribution of attribute values of a certain employee group having similar entitlements within the domain *International Operations*. While additional analyses show, that the orange value (domain *PATZ*) is occurring within this employee group rather normally (0.03% within this group vs 0.01% overall, the overall values are not displayed within the result matrix), only 1 out of 88 employees are attributed to the value *CFS*.



Count	Name	SA_SalesAndMarketing	PATZ	ExecutiveBoard	Breweries	IOSAfrica
499	B1 (499)	0.998				0.002
330	12 (330)	0.99	0.01			
151	06 (151)	0.96	0.03	0.01	0.01	
54	0E (54)	1				
49	04 (49)	1				

Figure 9: Entry of the result matrix with highlighting for the area Sales (screenshot is distorted)



Count	Name	CFS	IOSCentralLCS	ECM	IOSChina	PATZ
88	30 (88)	0.01(1)	0.88(77)	0.1(9)		0.01(1)

Figure 10: Entry of the result matrix with highlighting for the area International Operations (screenshot is distorted)

After the result interpretation and the exclusion of false positive results by IT experts, we arrived at a final set of 100 suspicious value assignments for the "IT domain" attribute out of the initially identified 184 outliers by the employee attribute correlation and 44 suspicious value assignments out of the initially identified 58 by the permission correlation. In total, 308 employees have been affected by these findings. We argue that manually discovering these errors would not have been possible. Only using automated correlation analysis provided by a tool-based prototype, a focused review by experts and the timely execution of the first three TAQM phases (and hence a structured attribute quality management for the IAM system) for FactComp has been made possible.

7.2. FinComp

In our second use case, we supported a company which operates in a highly-regulated environment during the improvement of their already existing IAM system. The organization manages 5.864 employees and 387 IT applications using a centralized IAM tool. Most of the application permissions are still managed manually, resulting in over-privileged employees. The access to building- and location-specific information managed via Microsoft Active Directory group memberships in specific was identified as error-prone. Reasons are, amongst others, the usage of complex group hierarchies within the Active Directory for handling the distribution of building-related information. As a result, we were asked to analyze the attribute quality of the employee attribute "building" which displays the building an employee is working in. Note, that in the following we only briefly describe each TAQM phase with a focus on highlighting differences and side effects that have not yet been discovered within our previously described use case. Table 10 shows the two important attributes for this use case. Within each phase different techniques to discover wrong attribute values for the "building" attribute are employed.

Phase I. : Personnel and organizational information together with account and permission data from 387 different applications was provided and imported in

Table 10: Relevant Attributes

Attribute	Attribute Description
City	Defines the city the employee is currently working at. Known to be correct.
Building	Defines the building the employee is currently located. As one building is assigned to exactly one city these two attributes should correlate for each account. Otherwise it would indicate a wrong attribute value. Known that attribute errors may occur.

our tool prototype. An initial attribute completeness check for the "building" attribute showed that only 92% of all employees are assigned to an attribute value, leaving 456 employees without a valid attribute value. These data errors (NULL-values) have been directly handed over for data cleansing and are excluded in the remainder.

Phase II. : During Phase II we firstly executed an account correlation with the two attributes above. We aimed at revealing correlations of specific application systems with certain buildings and created groups of employees according to the assignment of a user account within each of the imported IT applications. Note that we only considered applications that comprise between 50 and 750 user accounts in total for two reasons: Firstly, widely-used IT applications (such as the Active Directory) cannot be attributed to be only used within specific buildings. Secondly, applications with too few user accounts lead to a high number of false positive results. After a first execution and a subsequent reconfiguration of the applied thresholds (loop-back cycle of TAQM) we discovered 98 IT applications with conspicuous distributions of user accounts, affecting 587 employees' "building" attribute value.

Besides the account correlation we correlated the employees' "building" attribute with other existing master data attributes (analogous to Table 8 or Section 7.1). We, for instance, correlated the employee attribute "city" (describing the city the respective employee works in) with employees' "building" assignment. This additionally identified three suspicious value groups, affecting the "building" assignment of further three employees out of 5.864 which, looking at the data, work in a building which is not located within the city they work in.

Phase III. : During a detailed visual inspection regarding these results, we were able to discard a total of 409 suspicious employees together with IT experts of the company as false positive results. This resulted in remaining quality issues regarding 123 employees. We, for instance, employed the introduced grid visualization in order to confirm one finding of an employee in Singapore with

[5]Grid visualization is a component from the software *Nexis Controle*

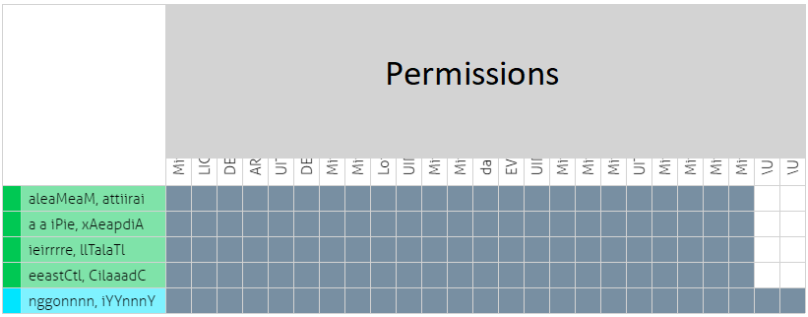


Figure 11: Grid visualization[5]filtered for a small department with colored employees based on their city attribute and having the building attribute value *Building_GER_2* (screenshot is distorted)

a wrong "building" attribute by coloring all employees based on their "city" attribute values.

The prototype displays all five employees having the pseudonymized building attribute value *Building_GER_2*. Note that all other employees not having this specific attribute have been excluded and are not shown. The remaining ones are coloured according to their "city" attribute (see Figure 11): The four green-coloured employees (city attribute value *Hamburg*) positively correlate with the mentioned building value. However, the only blue-colored employee with the city attribute *Singapore* but also with the building value *Building_GER_2* has two additional permissions, no other employee is assigned to (bottom right corner of the figure). The permissions' names are anonymized, however, in the original version they were clearly related to an Asian and Singapore region (e.g. calendar for region Asia). Therefore the "city" attribute is correct as the permission is indicating an Asian location while the value for "building" needs to be adjusted. One reason for such data quality errors might be inadequately enforced mover processes of employees who changed their work location and only received an update on their "city" attribute without adequately changing their "building" attribute.

7.3. AutoComp

The third use case covers a large, world-wide operating company in the automotive sector with more than 11,000 employees and more than 75,000 system entitlements within one application (see dataset #3 in Table 9). Within this use case we want to give an example how to exploit our findings by using the knowledge of experts within Phase IV to cleanse data quality issues. The company recently completed an HR-based project to introduce three new employee attributes ("job", "jobgroup", and "jobbox") for assigning access privileges according to employees' jobs within the organization. The project aimed at defining and assigning valid values for those attributes to all employees in a top-down manner, i.e. manually by experts for each department. During this

process, departmental managers were asked to provide their employees' job-, jobgroup-, and jobbox assignments based on a predefined list of valid values.

Phase I, II, and III. : After importing the provided data into our data storage we analyzed the attribute filling levels (job (96%), jobgroup (96%), and jobbox (71%)). Despite the fact that the attributes have a logical dependency (i.e. every job is assigned to a specific jobgroup and every jobgroup is assigned to a specific jobbox), they have been maintained in an independent manner. Consequently, we were able to find a total of 67 suspicious attribute value groups, covering 304 user accounts (out of 11,500 accounts) during Phase II. During Phase III, IT expert analysis confirmed these results to a large extent, for instance in a case where all except one employee with the same job value were in the same jobgroup.

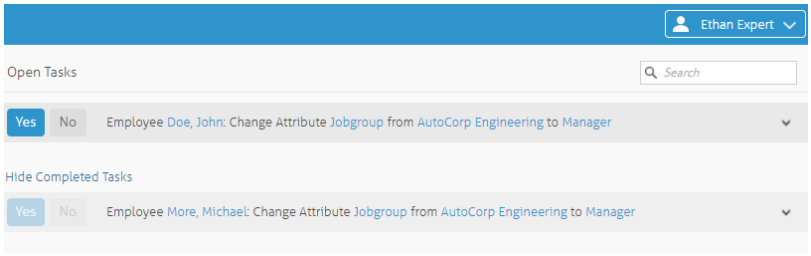


Figure 12: Verification[6] of anomalies via graphical user interface

Phase IV. : In order to cleanse the identified quality issues, our results were provided to responsible staff (e.g. the attribute owner) during Phase IV. We informed responsible IAM managers about the planned data cleansing process together with non-IT experts. In order to maximize user adoption, we used the existing graphical user interface for business experts of *Nexis Controle* (see Figure 12). The software's built-in delegation workflows already offered basic expert review processes which we extended in order to display TAQM results from Phase II and III to non-IT experts.

By using a simplified graphical user interface to review data quality issues, we foster the integration of organizational knowledge from non-IT staff. Potential stakeholders are departmental managers, attribute owners, entitlement owners, or other experienced employees. During TAQM execution we, for instance, automatically delegate findings from Phase III as review tasks to responsible experts. In the following, we present a simplified example of two exemplary tasks which have been delegated to an attribute owner. Via the graphical user interface he is asked to accept (e.g. cleanse) or decline (e.g. ignore) the attribute quality issues via a simple two-option button design. He can also optionally be

[6] Verification component is from the software *Nexis Controle*

allowed to browse the concerned entities for further information or review his previously completed tasks. In this example we asked "Ethan Expert" who is the owner of the attribute "jobgroup" to confirm two identified outliers and change the attribute value to *Manager* (i.e. a pre-calculated potentially correct value delivered by our employee attribute correlation during Phase II). The two employees were erroneously assigned the value *AutoComp Engineering* for their "jobgroup". Ethan Expert can, for instance, verify our findings by accepting both of the tasks to change the attribute values to the proposed new ones.

By using TAQM in combination with an expert-oriented simplified graphical user interface, companies are enabled to delegate different attribute quality tasks to non-IT experts. Additionally, expert decisions can be stored centrally for compliance or regulatory requirements. Note that, additionally to the already received positive feedback from the IAM managers, AutoComp currently is in the process of deciding about applying our approach during their next project phase.

8. Conclusion

The complexity and the number of challenges IAM has to tackle in modern companies is constantly rising. ABAC is one of the successors of RBAC and offers enough flexibility to overcome several access management challenges. However, the deployment of ABAC presents challenges itself that have not been addressed sufficiently yet by research. Essentially, a lack of attribute quality can lead to dysfunctional access control decisions and hence the existence of security vulnerabilities. Up to now there is no comprehensive attribute quality model ensuring a continuously high quality of attributes used within ABAC policies. To close this gap, we proposed TAQM, a structured approach for data quality management in IAM environments.

We initially derived a conceptual model for IAM, compared existing data quality approaches and analyzed their applicability for IAM. Subsequently, we defined TAQM as a process model and developed different tools to support the execution of each phase. In a last step we applied TAQM within three different IAM use cases. We were able to identify attribute value errors prior unnoticed and verify the suspicious quality issues in cooperation with company experts in real life projects.

After successful evaluation in real-world projects, we now plan to extend our research and monitor the long-term performance of TAQM. We want to analyze to which level the overall attribute quality increases over a longer period. Additionally, we aim at fine-tuning the various automation tools by integrating automated parameter configuration functionality for an easier application of TAQM. Lastly, we also want to investigate TAQM's extensibility towards other fields. One possibility would be the integration with identity behaviour analysis (e.g. answering questions like "are employees which are behaving similarly assigned to similar attribute values?"). Another option is applying previous expert decisions for an improved verification of anomalies. Past decisions could,

for instance, be used to verify identified outliers at run-time in order to increase the detection rate.

References

[1] SOX, Sarbanes-oxley act of 2002, pl 107-204, 116 stat 745 (2002).

[2] Basel Comittee on Banking Supervisions, Basel III: Int. framework for liquidity risk measurement, standards and monitoring.

[3] M. Hummer, M. Kunz, M. Netter, L. Fuchs, G. Pernul, Adaptive identity and access management—contextual data based policies, Journal on Information Security 2016 (1) (2016) 19.

[4] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman, Role-based access control models, Computer 29 (2) (1996) 38–47.

[5] L. Fuchs, G. Pernul, Hydro–hybrid development of roles, in: International Conference on Information Systems Security, Springer, 2008, pp. 287–302.

[6] V. C. Hu, D. F. Ferraiolo, D. R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone, Guide to attribute based access control (abac) definition and considerations, NIST Special Publication (2014) .

[7] N. K. Sharma, A. Joshi, Representing attribute based access control policies in owl, in: 10th International Conference on Semantic Computing, IEEE, 2016, pp. 333–336.

[8] V. Hu, D. F. Ferraiolo, D. R. Kuhn, R. N. Kacker, Y. Lei, Implementing and managing policy rules in attribute based access control, in: 16th International Conference on Information Reuse and Integration, IEEE, 2015, pp. 518–525.

[9] L. Fuchs, G. Pernul, R. S. Sandhu, Roles in information security–a survey and classification of the research area, Computers & Security 30 (8) (2011) 748–769.

[10] L. Fuchs, M. Kunz, G. Pernul, Role model optimization for secure role-based identity management, in: 22nd European Conference on Information Systems, AISeL, 2014.

[11] A. Elliott, S. Knight, Role explosion: Acknowledging the problem, in: 8th International Conference on Software Engineering Research and Practice, 2010, pp. 349–355.

[12] R. S. Sandhu, The authorization leap from rights to attributes: Maturation or chaos?, in: 17th ACM symposium on Access Control Models and Technologies, ACM, 2012, pp. 69–70.

[13] A. C. O'Connor, R. J. Loomis, Economic analysis of role-based access control, NIST, Gaithersburg, MD 2010.

[14] T. Priebe, W. Dobmeier, B. Muschall, G. Pernul, Abac–ein referenzmodell für attributbasierte zugriffskontrolle, in: Sicherheit, 2005, pp. 285–296.

[15] E. Yuan, J. Tong, Attributed based access control (abac) for web services, in: 9th International Conference on Web Services, IEEE, 2005.

[16] Z. Xu, S. D. Stoller, Mining attribute-based access control policies from rbac policies, in: 10th International Conference and Expo on Emerging Technologies for a Smarter World, IEEE, 2013.

[17] W. E. Deming, Out of the crisis, Cambridge University Press, 1986.

[18] R. Y. Wang, A product perspective on total data quality management, Communications of the ACM 41 (2) (1998) 58–65.

[19] C. Batini, M. Scannapieco, Data and information quality: Dimensions, principles and techniques, Springer, 2016.

[20] P. B. Crosby, Quality is free: The art of making quality certain, Signet, 1980.

[21] B. Aquilani, C. Silvestri, A. Ruggieri, C. Gatti, A systematic literature review on total quality management critical success factors and the identification of new avenues of research, The TQM Journal 29 (1) (2017) 184–213.

[22] J. P. Womack, D. T. Jones, D. Roos, Machine that changed the world, Simon and Schuster, 1990.

[23] P. S. Pande, R. P. Neuman, R. R. Cavanagh, The six sigma way, McGraw-Hill, 2000.

[24] M. Kaiser, M. Klier, B. Heinrich, How to measure data quality?–a metric-based approach, in: 28th International Conference on Information Systems, AISeL, 2007.

[25] H. Hinrichs, Datenqualitätsmanagement in Data Warehouse-Systemen, Springer, 2002.

[26] D. Ballou, R. Y. Wang, H. Pazer, G. K. Tayi, Modeling information manufacturing systems to determine information product quality, Management Science 44 (4) (1998) 462–484.

[27] A. K. Elmagarmid, P. G. Ipeirotis, V. S. Verykios, Duplicate record detection: A survey, IEEE Transactions on Knowledge and Data Engineering 19 (1) (2007) 1–16.

[28] D. Servos, S. L. Osborn, Current research and open problems in attribute-based access control, ACM Computing Surveys 49 (4) (2017) 65.

[29] J. Werner, C. M. Westphall, C. B. Westphall, Cloud identity management: A survey on privacy strategies, Computer Networks 122 (2017) 29–42.

[30] V. Hu, D. F. Ferraiolo, D. R. Kuhn, R. N. Kacker, Y. Lei, Implementing and managing policy rules in attribute based access control, in: Information Reuse and Integration (IRI), 2015 IEEE International Conference on, IEEE, 2015, pp. 518–525.

[31] A. R. Hevner, S. T. March, J. Park, S. Ram, Design science in information systems research, MIS quarterly 28 (1) (2004) 75–105.

[32] J. Sermersheim, Lightweight directory access protocol (ldap): The protocol (2006).
URL https://tools.ietf.org/html/rfc4511

[33] R. Philpott, N. Ragouzis, T. Wisniewski, E. G. Whitehead, H. Hinton, C. P. Cahill, J. Bradley, J. Hodges, J. Brennan, et al., Assertions and protocols for the oasis security assertion markup language (saml) v2. 0 errata 05 (2012).
URL https://www.oasis-open.org/committees/download.php/56776/sstc-saml-core-errata-2.0-wd-07.pdf

[34] D. Hardt, Rfc 6749 - the oauth 2.0 authorization framework (2012).
URL https://tools.ietf.org/html/rfc6749

[35] D. Rolls, Service provisioning markup language (spml) version 1.0 (2003).
URL https://www.oasis-open.org/committees/download.php/4137/os-pstc-spml-core-1.0.pdf

[36] T. Moses, extensible access control markup language (xacml) version 2.0 (2005).
URL http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[37] N. Naik, P. Jenkins, A secure mobile cloud identity: Criteria for effective identity and access management standards, in: 4th International Conference on Mobile Cloud Computing, Services, and Engineering, IEEE, 2016, pp. 89–90.

[38] P. E. Hunt, K. Grizzle, E. Wahlstroem, M. C., Rfc 7643 - system for cross-domain identity management: Core schema (2015).
URL https://tools.ietf.org/html/rfc7643

[39] S. Oh, S. Park, Task–role-based access control model, Information systems 28 (6) (2003) 533–562.

[40] A. A. E. Kalam, R. E. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, G. Trouessin, Organization based access control, in: 4th International Workshop on Policies for Distributed Systems and Networks, IEEE, 2003, pp. 120–131.

[41] P. J. Windley, Digital identity: Unmasking identity management architecture (IMA), O'Reilly Media, Inc., 2005.

[42] A. Pfitzmann, M. Hansen, A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management, technical report (2010).

[43] J. A. Long, C. E. Seko, A cyclic-hierarchical method for database data-quality evaluation and improvement, Information quality 1 (2005) 52–66.

[44] P. Falorsi, S. Pallara, A. Pavone, A. Alessandroni, E. Massella, M. Scannapieco, Improving the quality of toponymic data in the italian public administration, in: International Conference on Database Theory, Vol. 3, 2003.

[45] L. Fuchs, G. Pernul, Qualitätssicherung im identity- und access management, HMD Praxis der Wirtschaftsinformatik 50 (1) (2013) 88–97.

[46] Y. W. Lee, D. M. Strong, B. K. Kahn, R. Y. Wang, Aimq: A methodology for information quality assessment, Information & management 40 (2) (2002) 133–146.

[47] Y. Su, Z. Jin, A methodology for information quality assessment in the designing and manufacturing process of mechanical products, in: Information Quality Management: Theory and Applications, Idea Group Publishing Hershey, 2006, pp. 190–220.

[48] C. Batini, M. Scannapieco, Data quality: Concepts, methodologies and techniques, Springer, 2006.

[49] D. Loshin, Enterprise knowledge management: The data quality approach, Morgan Kaufmann, 2001.

[50] M. Scannapieco, A. Virgillito, C. Marchetti, M. Mecella, R. Baldoni, The daquincis architecture: A platform for exchanging and improving data quality in cooperative information systems, Information systems 29 (7) (2004) 551–582.

[51] L. L. Pipino, Y. W. Lee, R. Y. Wang, Data quality assessment, Communications of the ACM 45 (4) (2002) 211–218.

[52] M. A. Jeusfeld, C. Quix, M. Jarke, Design and analysis of quality information for data warehouses, in: 17th International Conference on Conceptual Modeling, Springer, 1998, pp. 349–362.

[53] M. J. Eppler, P. Muenzenmayer, Measuring information quality in the web context: A survey of state-of-the-art instruments and an application methodology, in: 7th International Conference on Information Quality, 2002, pp. 187–196.

[54] F. De Amicis, C. Batini, A methodology for data quality assessment on financial data, Studies in Communication Sciences 4 (2) (2004) 115–137.

[55] L. P. English, Improving data warehouse and business information quality, J. Wiley & Sons, 1999.

[56] D. Brossard, G. Gebel, M. Berg, A systematic approach to implementing abac, in: Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control, ACM, 2017, pp. 53–59.

[57] S. Bhatt, F. Patwa, R. Sandhu, Abac with group attributes and attribute hierarchies utilizing the policy machine, in: Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control, ACM, 2017, pp. 17–28.

[58] I. Indu, P. R. Anand, V. Bhaskar, Identity and access management in cloud environment: Mechanisms and challenges, Engineering Science and Technology, an International Journal.

[59] U. Habiba, R. Masood, M. A. Shibli, M. A. Niazi, Cloud identity management security issues & solutions: a taxonomy, Complex Adaptive Systems Modeling 2 (1) (2014) 5.

[60] R. Likert, A technique for the measurement of attitudes, Archives of psychology 22 (140) (1932) .

[61] B. Heinrich, M. Kaiser, M. Klier, Does the eu insurance mediation directive help to improve data quality? - a metric-based analysis, in: 16th European Conference on Information Systems, AISeL, 2008.

[62] M. Kunz, L. Fuchs, M. Hummer, G. Pernul, Introducing dynamic identity and access management in organizations, in: 11th International Conference on Information Systems Security, 2015, pp. 139–158.

[63] L. Fuchs, Methodology for Hybrid Role Development, Vol. 69, Eul Verlag, 2010.

[64] S. Meier, L. Fuchs, G. Pernul, Managing the access grid - a process view to minimize insider misuse risks, in: 11th International Conference on Wirtschaftsinformatik, 2013, pp. 1051–1065.

[65] J. Venable, J. Pries-Heje, R. Baskerville, A comprehensive framework for evaluation in design science research, in: 7th International Conference on Design Science Research in Information Systems, Springer, 2012, pp. 423–438.

2 Optimization of Access Control Policies

Current status:	Published
Journal:	Journal of Information Security and Applications, Volume 70, November 2022
Date of acceptance:	31.07.2022
Full citation:	Sascha Kern, Thomas Baumer, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Optimization of access control policies. *Journal of Information Security and Applications (JISA)*, 70:103301, 2022.
Authors contributions:	Sascha Kern 45% Thomas Baumer 30% Sebastian Groll 10% Ludwig Fuchs 5% Günther Pernul 10%

Conference Description: Journal of Information Security and Applications (JISA) focuses on the original research and practice-driven applications with relevance to information security and applications. JISA provides a common linkage between a vibrant scientific and research community and industry professionals by offering a clear view on modern problems and challenges in information security, as well as identifying promising scientific and "best-practice" solutions. JISA issues offer a balance between original research work and innovative industrial approaches by internationally renowned information security experts and researchers.

Optimization of Access Control Policies

Sascha Kern[a,*], Thomas Baumer[a], Sebastian Groll[a,b], Ludwig Fuchs[a] and Günther Pernul[b]

[a]Nexis GmbH, Franz-Mayer-Straße 1, Regensburg, 93053, Bavaria, Germany
[b]University of Regensburg, Universitätsstraße 31, Regensburg, 93053, Bavaria, Germany

ARTICLE INFO

Keywords:
Access Management
Data Quality
Policy Optimization
Policy Maintenance
Role-Based Access Control
Attribute-Based Access Control

ABSTRACT

Organizations undertake complex and costly projects to model high-quality Access Control Policies (ACPs). Once built, these policies must be maintained and managed in an ongoing process to keep their quality high. Insufficient maintenance leads to inaccurate authorization decisions and increases the policies' administrative effort and susceptibility to errors. While the initial modelling of ACPs has received significant research interest, their optimization is not yet covered as broadly. This work provides a theoretical foundation for ACP quality and its optimization. Furthermore, it analyzes how existing research addresses optimization of ACPs with regard to six crucial optimization dimensions. It presents a structured literature survey tracing these optimization dimensions, the contributed research artifact and data requirements. Building on this literature catalogue, this work elaborates on inaccuracies for user permission assignments, data availability, minimal perturbation and recommendation-based optimization.

1. Introduction

The organizational structures and IT infrastructures of modern companies are subject to constant change. Routine operations like departmental changes of employees, changing responsibilities or the integration of new application systems into the IT landscape require an adaptation of IT security configurations. This includes updating Access Control Policies (ACPs), machine-processable rules that define authorizations and can be evaluated in a fully automated manner to determine which accesses an employee is allowed to make (Samarati and de Vimercati, 2001). Due to changing environmental conditions, ACPs that were once of high quality lose accuracy over time (Xia et al., 2014; Hu et al., 2011). Moreover, ACPs proliferate over time, as policy administrators may over-grant access to conform with immediate business needs (Xu et al., 2017; Xiang et al., 2019) or update policies in an erroneous or non-optimal way. Besides hard errors, ACP proliferation leads to lower comprehensibility and maintainability, which increases the ACPs' administrative cost and their proneness for further errors (Bauer et al., 2009; Beckerle and Martucci, 2013).

Incorrect or overly permissive access decisions leave companies vulnerable to insider threats. A malicious or careless insider can harm an organization severely, with consequences spanning from unintentional incidents to sabotage, fraud or espionage (Tsiostas et al., 2020). In contrast, overly restrictive access decisions prevent employees from doing their work, leading to costly interruptions in operations and task backlogs. Recent studies estimate that the average annual cost of insider threats for companies reach $11.45 million in 2020 (Gilbert, 2021; Tsiostas et al., 2020). The implementation of effective Identity and Access Management (IAM) measures, which follow the principle of least privilege (Horne, 2011), is hence mandated by major regulatory frameworks and IT security standards[1].

Maintenance measures, which optimize the quality of ACPs in a continuous manner, are a fundamental requirement for providing an accurate level of security with reasonable administrative effort over a longer period of time, and for maintaining the investment made in the initial modelling of high quality ACPs (Molloy et al., 2010; Parkinson et al., 2020; Kunz et al., 2015a). As the initial modeling of ACP sets with high quality requires high time and financial effort (Jaferian et al., 2014), maintenance processes aim to improve the quality of an ACP set by applying updates that leave the existing state intact. ACP maintenance is commonly approached in two types of processes: First, access reviews are a process where responsible humans (such as a department head) review ACPs for entities in their responsibility (for example roles assigned to their employees) and try to find and rectify inaccuracies. The effectiveness of access reviews is limited, since reviewers have to check large amounts of data in a largely manual process and have limited information to make a qualified decision (Groll et al., 2021; Pan et al., 2018; Jaferian et al., 2014). Second, ACP refinement processes aim to improve the quality of an ACP set by updating it in a (semi-)automated manner without deconstructing it (Xia et al., 2014). Both approaches will be considered in the course of this work.

This work contributes to an optimization of ACPs by addressing the following **research question**: *Which methods for the optimization of roles and Attribute-Based Access Control (ABAC) policies are present, and what are their*

*Corresponding author: S. Kern
sascha.kern@wiwi.uni-regensburg.de (S. Kern)
https://nexis-secure.com/ (L. Fuchs);
https://www.uni-regensburg.de/wirtschaftswissenschaften/wi-pernul/startseite/index.html (G. Pernul)
ORCID(s): 0000-0002-7082-000X (S. Kern); 0000-0003-0157-3057 (T. Baumer); 0000-0003-1338-9003 (G. Pernul)

[1]Such as the Sarbanes-Oxley Act (One Hundred Seventh Congress of the United States of America, 2002), the Basel Accords (Basel Committee on Banking Supervision), the European General Data Privacy Regulation (The European Parliament and the Council of the European Union, 2016), the ISO 27000 standards (International Organization for Standardization, 2013), or the BSI Grundschutz (Bundesamt für Sicherheit in der Informationstechnik, 2019).

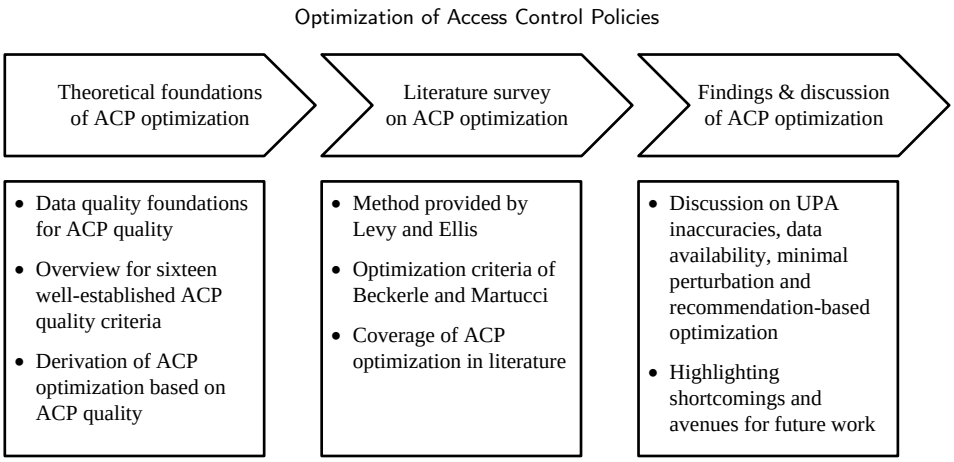


Figure 1: General approach of the study

advantages and limitations?. The contribution of this work is threefold: (i) We provide a definition of ACP quality as an instance of the data quality concept and supplement it with a collection of 16 ACP properties that are repeatedly used in literature to determine ACP quality. Building on this, we define ACP optimization as an improvement of the quality of existing ACPs. (ii) We conduct a structured literature survey based on the methodology proposed by Levy and Ellis (2006). Our scope is set for scientific publications describing means for optimization of ACPs that satisfy at least one of the six optimization objectives of Beckerle and Martucci (2013). After obtaining a literature catalogue for publications on optimization of ACPs, we categorize and analyze the literature catalogue. (iii) Finally, we build on findings from the literature survey and discuss important aspects of ACP optimization in more detail. At first, we discuss prototypical approaches to identify User Permission Assignment (UPA) inaccuracies, their advantages and disadvantages and their data requirements. Subsequently, we discuss the availability of three classes of data on which ACP optimization methods commonly rely in order to analyze the consequence of these data requirements. We then discuss the concepts of minimal perturbation and recommendation-based optimization and their addressing in existing literature. In addition, current shortcomings and research gaps are identified and avenues for future work can be highlighted. The discussion is presented in section 5. Figure 1 gives an overview of the general approach.

This work is focused on the Access Control Models (ACMs) Role-Based Access Control (RBAC) (Sandhu, 1998) and ABAC (Hu et al., 2014) since these are the most common ACMs. The standard RBAC model defines ACPs in the form of roles, which are bundles of permissions that can be assigned to subjects (e.g. employees) in a well-organized way. RBAC also offers the definition of constraints, which are statements that express negative authorizations, i.e. authorizations that must not be granted by the role set. ABAC ACPs in contrast are modeled as (dynamic) policies which make authorization decisions based on attributes of subjects (e.g. employees), objects (permissions) or the execution environment (e.g. the execution time). The term ACP hence refers to (constrained) RBAC roles as well as to ABAC policies (Samarati and de Vimercati, 2001). The Organization for the Advancement of Structured Information Standards (OASIS) defined the eXtensible Access Control Markup Language (XACML) standard, which provides a notation for expressing ABAC policies in an XML-based format as well as a reference architecture for a policy evaluation mechanism (Godik and Moses, 2003). XACML is the most important technical standard for ABAC and is explicitly addressed in many ABAC-related publications. Note that ABAC was often proclaimed as the successor of RBAC[2] since ABAC has the descriptive strength to express RBAC along with other attribute-based policies. E.g. Cheminod et al. (2018) showcase this behavior for an industrial use case. However, ABAC has still not reached the maturity of RBAC and has difficulties with practical adoption (Servos and Osborn, 2017; Puchta et al., 2019). Most issues of ABAC are rooted in the raised flexibility of attributes and ABAC policies which backfire as increased complexity and thus are less comprehensive for administrators and policy engineers (Servos and Osborn, 2017). Additionally, spill-over effects from RBAC to ABAC and vice versa can be observed in literature (Gardiyawasam Pussewalage and Oleshchuk, 2017; Qi et al., 2018; Nazerian et al., 2019). Further well-known ACMs, like Discretionary Access Control (DAC) and Mandatory Access Control (MAC) (Samarati and de Vimercati, 2001), or very recent approaches like Attribute-aware Relationship-Based Access Control (AReBAC) (Cheng et al., 2014; Chakraborty and Sandhu, 2021b,a) are not in the scope of this work.

The remainder of this paper is structured as follows. Section 2 presents work that is related to the quality of ACPs and their optimization. Section 3.2 defines ACP quality and ACP optimization and presents common quality criteria.

[2] In 2013 Gartner stated that "*by 2020 70% of enterprises will use attribute based access control [...] as dominant mechanism to protect critical assets [...]*" https://www.gartner.com/en/documents/2607617

Section 4 presents a literature survey on optimization of ACPs. Building on the findings of the survey, further aspects of ACP optimization are discussed in section 5. Finally, section 6 sums up the results and concludes this work.

2. Related Work

While existing studies present a comprehensive picture of RBAC and ABAC research and open challenges (Fuchs et al., 2011; Servos and Osborn, 2017), to the best of our knowledge no literature study has examined the more specific field of ACP optimization yet. Since publications that contribute to this field directly are presented and analyzed in the literature survey in chapter 4, this chapter serves to present research areas that are closely related to ACP quality optimization.

The assessment of ACP quality is a fundamental building block for their optimization. Since ACP quality comprises many dimensions, the scientific literature is heterogeneous and proposes many ways to assess ACP quality. This includes the definition of quality metrics (Jabal et al., 2019; Beckerle and Martucci, 2013) and distinct research realms that aim at individual quality-related objectives. Examples are ACP anomaly analysis, which aims at the identification of conflicts and redundancies, or XACML evaluation runtime analysis, which aims to find causes for a slow evaluation runtime. Moreover, ACP quality research is concerned with the question of how the quality of ACPs develops in real environments and which structural reasons are responsible for this. Researchers have documented that the quality of ACPs gradually deteriorates without targeted countermeasures, and have identified structural causes such as structurally determined over-granting (Xiang et al., 2019) or the role explosion problem (Elliott and Knight, 2010).

The initial creation of ACPs with high quality is addressed by ACP modelling approaches. Existing approaches differ greatly in terms of modelling objectives and their definitions of *optimal* ACPs (Kunz et al., 2015b; Mitra et al., 2016). Existing proposals typically focus on policy mining, i.e. the automatic generation of new ACPs based on *existing* authorization structures, or policy engineering, i.e. the (mostly manual) definition of new ACPs based on an *ideal* authorization structure. Hybrid approaches aim to utilize the advantages of mining and engineering (Cotrini et al., 2018; Fuchs and Pernul, 2008). While the initial creation of high-quality ACPs does not constitute an optimization, some RBAC or ABAC mining algorithms provide a maintenance mode that works on existing ACPs and were included in the literature survey. The repeated re-generation of an entire policy set is an alternative approach to keep ACPs up-to-date (Calo et al., 2019; Verma et al., 2019). It is based on the assumptions that policies remain unchanged during their whole life cycle and can be generated on a sufficiently high quality level without any human interaction. Instead of maintaining existing ACPs, this approach aims to ensure a sufficient quality be re-generating all policies from scratch on a regular basis.

3. Theoretical Foundations of ACP Optimization

Before presenting the methodology and findings of the survey, this chapter introduces the theoretical foundations for ACP quality and its optimization. Section 3.1 demonstrates that ACP quality is an instance of the data quality concept as defined by Wand and Wang (1996) and provides a definition of ACP quality based on it. To complement this definition, section 3.2 presents a collection of 16 well-established ACP quality criteria. Building on these preliminaries, section 3.3 concludes the theoretical foundation with a definition of ACP optimization.

3.1. Quality of Access Control Policies

Present research on ACP quality shows similarities to data quality research: There is consensus in both fields that quality is a multidimensional concept and cannot simply be assessed as universally *good* or *bad*. As in the field of data quality research, IAM research also applies different criteria to assess the quality of ACPs. While some of them only make sense in the context of ACPs (like the grade of automation or evaluation runtime), others are equivalent to data quality dimensions that are well-established outside the research realm of IAM (e.g. accuracy, understandability, completeness or redundancy). To the best of our knowledge, present research did yet not formally show that ACP quality constitutes an instance of the data quality concept. In this section, we show that the widely acknowledged data quality model by Wand and Wang (1996) is applicable to ACPs and provide a definition of ACP quality based on it.

The basis for an organization's access control decisions is its security policy (Sandhu and Samarati, 1994; Samarati and de Vimercati, 2001). A security policy is a collection of (often informal) requirements that define the authorizations of an organization. One of the most commonly quoted security policy requirements is the principle of least privilege, which states that a subject should not inherit more permissions than it needs to perform its tasks (Horne, 2011). A security policy may also comprise requirements that are not directly related to the organization's security, for example to fulfill organizational or regulatory needs. One example are Segregation of Duty (SoD) requirements, which define mutually exclusive authorizations and are commonly employed to avoid conflicts of interest. Access Control Policies (ACPs) are machine-processable rules that can be evaluated in a fully automated manner to determine which accesses a subject is allowed to make (Samarati and de Vimercati, 2001). Following these established definitions, ACPs are a data representation of the authorizations that are specified by an organization's security policy.

The data quality model by Wand and Wang (1996) defines an information system as an entity which exists parallel to a real-world system. The data stored in the information system is a representation of a perception of the real-world system. Through interpretation of this data, a user perceives a view of the real-world system as inferred from the information system. The process of creating data

that represents real-world entities is called the representation transformation. The process of creating an interpretation of the representational data which resembles the original real-world entities is called interpretation transformation. If both, representation transformation and interpretation transformation, work correctly, the view of the real-world system as inferred from the information system is identical to the view of the real-world system gained from direct observation. Any disparity in between these views represents a data deficiency. The authors define four assumptions which need to be met for the model to be applicable:

(i) *The Representation Assumption: An information system is a representation of a real-world system as perceived by users.* The authorizations defined by an organization's security policy exist outside the information system and are hence (abstract) real-world entities. The ACPs that express them are consequently a data representation of real-world-entities, and the information system that stores them is a representation of a real-world system.

(ii) *The Interpretation Assumption: An information system is built for use by the user whose view of the real-world system is captured in the design of the system.* Wand and Wang explain that this assumption serves to ensure that the interpretation transformation (i.e. the process of transforming data back into perceivable real-world entities) will be able to map the data representation back to the original real-world entities. In the instance of ACPs, the design of the information system equals the view of the information system, because the representation transformation and the interpretation transformation are based on the same access control model: Any set of authorizations can be represented as a user-permission matrix. Both access control models in scope, RBAC and ABAC, allow to define an ACP set for every possible user-permission matrix, that will be mapped back the the exact same user-permission matrix.

(iii) *The Inference Assumption: The information system can create a perceptible representation from which the user can infer a view of the real-world system as represented in the information system.* Since every ACP set can be represented (and also visualized) as a user-permission matrix, an information system can always infer a view of the original authorizations that are represented by the ACP set.

(iv) *The Internal View Assumption: Issues related to the external view such as why the data are needed and how they are used are not part of the model.* This assumption is self-fulfilling as it is merely states that the model does not deal with external issues.

By showing that ACPs fulfill these four assumptions, we show that the data quality model by Wand and Wang (1996) is applicable to ACPs and that the concept of ACP quality constitutes an instance of the data quality concept. Present research widely agrees that the quality of data is best described as its "fitness for use" (Tayi and Ballou, 1998). In accordance with this definition, we define the quality of ACPs as their fitness for use with regard to one or more quality dimensions that reflect the application context of access control.

3.2. Established Quality Criteria

Present research applies many different criteria to evaluate the quality of ACPs, many of which include a concept of optimal quality. Beckerle and Martucci (2013) developed six criteria to determine well-usable ACPs sets and developed metrics to quantify these. Both Kunz et al. (2015b) and Mitra et al. (2016) present surveys on role mining approaches and point out objectives that role mining algorithms apply to achieve high-quality roles. Jabal et al. (2019) define a list of policy analysis criteria with implications to the policy quality. Besides that, a large number of both RBAC and ABAC related publications define quality criteria "on the fly" and often also define metrics for these quality criteria to approach a particular objective at hand.

In order to complement the definition of ACP quality, the remainder of this section presents a collection of properties of ACPs that are commonly applied in existing literature to evaluate the quality of ACPs. The collection comprises 16 properties, 14 of which have an optimum in terms of ACP quality. The remaining two criteria (usage and relevance) are often used in the context of quality assessment, but are not an expression of ACP quality themselves. Furthermore, seven of the presented criteria affect the evaluation of ACPs by the access control mechanism directly, while the remaining nine only implicitly affect their correct evaluation through factors such as error-proneness during policy administration or maintenance efficiency. Please note that this is not a complete list, as creating a full list of established quality criteria would require a structured, reproducible literature survey on its own. Individual quality criteria may be positively or negatively correlated, or not influence each other at all: For example, adding new rules to an ABAC policy will likely increase its UPA coverage and bring it closer to being complete. At the same time, the complexity of the policy is increased, which suggests a negative correlation between the objectives of minimal complexity and maximal completeness. Kunz et al. (2015b) present a dependency analysis of quality criteria applied during role mining. Despite that, to the best of our knowledge, the interaction of ACP quality criteria has not been analyzed by scientific research yet. Table 1 provides an overview of all 16 properties, their optima and whether or not the criteria affect the ACP evaluation directly.

Accuracy is the most important quality dimension for ACPs as it expresses the effective correctness of their access control decisions. It is hence directly related to the correct evaluation of the policies. The accuracy of an ACP set defines, how accurately it represents the authorizations defined by the security policy[3] (Beckerle and Martucci, 2013). There are two types of errors that decrease the accuracy of an ACP set: If an ACP set grants excessive UPAs, subjects inherit more permissions than they require. In contrast, missing UPAs mean that subjects require particular permissions, but are not

[3] Beckerle and Martucci (2013) refer to the Security Policy as "Access Control Policy". To avoid confusion, this work uses the more common term "Security Policy". The term "Access Control Policy" refers to the data representation of the authorizations defined by the security policy, which is subject to optimization.

Table 1
Common quality-related ACP properties

Property	Optimum	Affects Evaluation
Accuracy	max	yes
Excessive UPAs	min	yes
Missing UPAs	min	yes
Maintainability	max	no
Understandability	max	no
Sem. meaningfulness	max	no
Complexity	min	no
Redundancy	min	no
Conflicts	min	yes
Grade of automation	max	yes
Evaluation runtime	min	yes
Similarity to opt. state	max	no
Risk	min	no
Completeness	max	yes
Usage	ambiguous	no
Relevance	ambiguous	no

granted them by the ACP set. The challenge in identifying inaccurate UPAs lies in determining which UPAs *should* be granted. If the entire set of correct UPAs was known, an ACP set could be optimized for perfect accuracy in a fully automated manner.

Excessive UPAs are a violation of the principle of least privilege and can cause a security vulnerability. An example of how harm can be done by excessive authorizations is when a hospital employee publishes sensitive patient data, either accidentally or in malicious intent. Excessive authorizations can also be abused by external attackers, for example, if an authorized employee is blackmailed or his or her user account is hijacked. The minimization of excessive UPAs is the primary objective when modelling and administering ACPs and is required to enable an acceptable level of security. Finding excessive UPAs however poses a greater challenge, and over-allocated privileges often go unnoticed until they are misused for a malicious act. For this reason, excessive UPAs tend to accumulate over time, making targeted countermeasures necessary (Fuchs et al., 2014). The amount of excessive UPAs that are granted by an ACP set is hence a crucial indicator for its quality.

Missing UPAs keep subjects from doing their work and hence conflict with business continuity. For example, a company's supply chain could suffer outages because an employee lacks the authorization to post a goods receipt in the enterprise resource planning system. Missing UPAs have a higher visibility than excessive UPAs because their damage occurs relatively quickly: A subject who has been wrongly deprived of an entitlement can immediately thereafter no longer perform a certain task. The impact can be substantial since this can include very basic privileges, such as authorization to enter an organization's premises or to log into their work station.

Maintainability describes how well an ACP set can be administered and kept up-to-date (Benedetti and Mori, 2019; Cheng et al., 2019). Low maintainability makes an ACP set prone to errors and leads to a higher administration effort. Increasing an ACP set's maintainability is hence a prime objective of ACP optimization. The maintainability of an ACP set is influenced by several properties including its understandability, its complexity, its redundancy or the amount of conflicts that it contains, which can be assessed and optimized individually. Tool-supported administration of ACP also helps early on in keeping desired properties up-to-date (Seifermann et al., 2022).

Understandability expresses how well an ACP set can be understood by humans. It is closely related to maintainability and is often cited together (Cheng et al., 2019; Hummer et al., 2015). A cryptic ACP set that is hard to understand is also hard to administer or maintain: For example, a policy administrator could misunderstand the meaning of an ABAC policy and make erroneous changes that cause inaccuracies. Alternatively, an administrator could decide not to make changes at all to a policy that he or she does not understand, leading to fast obsolescence of that policy. Maintaining a good understandability is hence considered a prime challenge by several authors (Kunz et al., 2019; Meier et al., 2013). A key factor for understandable ACPs is semantic meaningfulness. Moreover, several authors proposed approaches for visualizing ACPs, which aim to improve the understandability of an ACP set without applying any changes to it (Puchta et al., 2019).

Semantic meaningfulness means that ACPs represent a human-understandable real-world concept. It is often argued to be crucial for ACP understandability (Xu, 2014; Molloy et al., 2010). Since semantic concepts can be described with attribute values, the semantic meaningfulness of an ACP can be assessed by measuring its accordance with semantically meaningful attributes (Xu, 2014). For example, a role that can be described as "This role grants all permissions that are required for all software developers" would accord to 100% to the value "software developer" of the semantically meaningful employee attribute "job title" and hence has a very high semantic meaning. Note that this definition does not require the ACP to be defined based on attributes itself. Moreover, an attribute-based ACP does not automatically have a higher semantic meaning than a role, since it can define a long list of permitted or denied UPAs which share little or no semantic meaning.

Complexity expresses the amount of elements that an ACP set consists of. For example, an ABAC policy that comprises 200 statements is likely more complex than one that contains only 5 statements. Similarly, a set of 50 roles with hierarchies among them and numerous permission assignments is likely more complex than a set of 5 roles with few permission assignments and no role hierarchies. Low complexity improves the maintainability of an ACP set and reduces the computational effort required for its evaluation. Existing

research assesses the complexity of an ACP set in numerous ways, including the amount of roles and ABAC policies contained in it, the size of roles and ABAC policies and more specific measurements. The most generic definition of ACP complexity is the Weighted Structural Complexity (WSC), which is a weighted sum of all elements defined by the underlying ACM (Molloy et al., 2008; Xu, 2014). The complexity of an ACP set is among the most commonly cited quality indicators. Unlike its understandability or semantic meaningfulness, the complexity of an ACP set can be quantified objectively without requiring any further data.

Redundancy occurs if an ACP set defines positive or negative authorizations more than once. For example, an employee could inherit the permission to close customer requests in a ticketing system twice because he has the role "customer support employee" and the role "administrator of ticketing system". Similarly, a redundant negative authorization could occur for a bank employee if an ABAC policy defined that no bank employee who serves private customers may approve lending, and that no employee who is still in training may approve lending. Redundancy leads to an unnecessary bloating of ACP complexity. Moreover, ACP redundancy is a possible cause for administration errors, as a redundantly defined positive or negative authorization must be removed more than once for the change to become effective. If one of the redundant definitions is overlooked in the process, the ACP set obtains an inaccuracy. The impact can be substantial, for example when an emergency permission revocation process (i.e. a process where a subject is immediately stripped of all permissions, for example because the digital identity was stolen) fails because a redundant permission assignment is overlooked.

Conflicts exist within an ACP set if it defines both positive and negative authorizations for the same user-permission pairs. This means that a particular permission is both allowed and forbidden for the same user. For example, an ABAC policy could state that IT administrators have file access to an application server, while at the same time denying access to personnel data files for anyone outside the human resource department. While a static conflict is present within the effective UPAs that an ACP set realizes at a given time, dynamic conflicts are *potential* conflicts, i.e. conflicts that could arise due to the dynamic nature of the ACPs, but did not necessarily generate a contradiction yet (Dunlop et al., 2003). While ABAC or XACML ACP sets define both positive and negative authorizations by default, RBAC ACP sets can only contain conflicts if used with RBAC constraints (Sandhu, 1998) since unconstrained roles define only positive authorizations. If a conflict exists within an ACP set, the access control mechanism must resolve it in order to make an unambiguous authorization decision. This is achieved by applying a conflict resolution strategy which defines how to make authorization decisions if a conflict occurs. The XACML standard defines basic conflict resolution strategies (Moses, 2005) and many more sophisticated conflict resolution algorithms were proposed by researchers. However, conflict resolution only aims to enable the access control mechanism to make a deterministic decision *despite* the presence of conflicts. It does not update the ACP set to remove the conflict and hence does not constitute a quality optimization. Since conflicts make an ACP set's authorizations ambiguous, they are a possible cause for inaccuracies and reduce the ACP set's understandability. Moreover, real-time conflict resolution reduces the evaluation performance of the ACP set.

The **grade of automation** of an ACP set determines, to which extent the authorizations defined by it adapt to new situations dynamically without requiring manual updates. For example, an ABAC policy that defines authorizations based on an employee's department affiliation requires no updating of a policy definition if an employee moves into another department, since the employee's department attribute value would change, thus leading to an updated result in the evaluation of subsequent authorization requests. Attribute-based ACPs inherently offer the possibility of dynamic rule definitions, since changes in referenced attribute values also change the authorization decisions resulting from ACPs evaluation. The standard RBAC model (Sandhu, 1998) in contrast is static and cannot update authorizations automatically unless it is extended with an automation mechanism (such as Kern and Walhorn (2005); Al-Kahtani and Sandhu (2004); Aftab et al. (2015)). While ACP automation is is directly related to the policy evaluation, it is also a critical factor in ACP maintenance as it reduces administrative effort and prevents excessive and missing UPAs before they occur (Fuchs et al., 2014; Kunz et al., 2015a). However, it cannot make ACP maintenance obsolete, since an automation mechanism operates with a limited scope, and dynamic ACP definitions can out-date or be erroneous like static ones (cmp. section 4.3.6). Additionally, automation also eases other IAM processes like policy refinement, policy verification or conflict resolution (Cheminod et al., 2017, 2019).

The **evaluation runtime** of an ACP set determines how quickly it can be loaded and evaluated by an access control mechanism to answer an authorization request (Turkmen and Crispo, 2008; Miseldine, 2008). A sufficiently low evaluation runtime is critical if ACPs must be evaluated in real-time, since a pending authorization decision is a performance bottleneck for all relying application systems. As a result, users could spend considerable time waiting for simple button clicks to be executed, or performance-critical operations in an organization's IT infrastructure such as large data processing tasks could pile up. The real-time evaluation of authorization requests by a central access control mechanism is a requirement specified in the OASIS XACML reference architecture (Hu et al., 2014). The evaluation runtime of an ACP set can be influenced by many factors, like the amount of contained ACPs, or the amount, size, order of the rules that an ACP contains or the algorithm used (Marouf et al., 2011; Deng et al., 2021).

Similarity expresses how similar one ACP set is compared to another. For example, a set of 10 department roles is likely more similar to a set of 5 department roles and 5 roles that represent an employee's job title than it is to a set of 50 roles which represent employees' job titles. Similarity is often used as a quality criterion by measuring how similar the assessed ACP set is to another ACP set which is considered optimal (Narouei and Takabi, 2019; Hadj et al., 2017). By maximizing its similarity to the optimal ACP set, a new ACP set can be generated that resembles the structure of an existing, well-structured ACP set, but includes the results of the newly applied generation algorithm. Moreover, similarity can be used to estimate the amount of updates that is required to migrate from one ACP set to another (Vaidya et al., 2008). By applying high similarity to the original ACP set as an objective, an ACP optimization method can produce updates with *minimal perturbation*, which is expected to reduce the administrative effort required for implementing the changes.

Risk determines the impact of excessively assigned permissions for an ACP or an ACP set. For example, a policy that grants subjects full access to a banking system is more risky than a policy that grants subjects access to a WiFi hotspot. The risk of an ACP reflects the aggregated risk of the permissions that are permitted by it. ACP modelling or optimization methods can use risk minimization as an objective to reduce the impact of excessive permission assignments (Jin et al., 2016; Dos Santos et al., 2014). High risk is also an indicator for high maintenance priority and can serve as context information for policy engineers and reviewers, based on the assumption that a high risk value suggests a more restrictive handling than a low one (Fuchs et al., 2014).

Completeness determines the amount of UPAs that are covered by an ACP set. Examples for incomplete UPA sets could be a set of roles which contain only a subset of the permissions managed by an organization, or an ABAC policy that only contains attribute definitions to make authorization decisions for a subset of the requested subjects. If an ACP set does not cover UPAs, an access control mechanism is unable to determine an access control decision for the corresponding user-permission pair during policy evaluation (except for a standard fallback decision). The criterion of completeness is also often used in ACP modelling to determine to which extent a newly modelled ACP set expresses the UPAs defined by an input state (Kunz et al., 2015b). The term coverage is closely related and often used interchangeably, but does not necessarily contain a quality indication. The coverage of an ACP is an indicator for its relevance, based on the assumption that a policy that defines many authorization decisions is more important than a policy that defines few ones.

Usage determines how often an ACP was invoked, i.e. how often the authorizations defined by an ACP were requested and executed. For example, an employee in the goods receiving department of a company might use the authorization to debit a delivery every day. An example of an infrequently used UPA could be that a back office employee needs to submit a balance sheet only once a year. If an UPA is never executed, this is an indicator that it is not needed, i.e. it is excessively assigned according to the principle of least privilege. The usage of an ACP can be reconstructed via access logs (as defined in section 4.2.3), by counting the invocations of the UPAs that are covered by the ACP over a defined timespan. ACP usage can be used to determine UPA inaccuracies (cmp. section 5.1.2) and is hence an important tool for ACP quality assessment. Moreover, the usage of an ACP is an indicator of its relevance, assuming that an ACP that is often invoked is more important than one that is scarcely invoked (Pan et al., 2018; Hadj et al., 2018a).

Relevance expresses how strongly an ACP influences authorization decisions in productive operations. For example, a basic role that allows every employee to access their work stations is likely more relevant than a specialist role that allows few employees to create a new email distribution list. ACP relevance is commonly analyzed to determine priorities when assessing or optimizing ACP quality. In case of conflicting policies, relevance can be used to prioritize a policy that should overrule another (cmp. section 4.3.5). The relevance of a policy can also be an indicator of timeliness (Bauer et al., 2011), and ACPs with low relevance can be interpreted as a security risk since they are likely to be over-permissive (Jabal et al., 2019). Moreover, policies with higher relevance can be maintained with higher priority in order to improve the effectiveness of ACP maintenance (Hadj et al., 2019). The properties *Risk*, *Completeness* and *Usage* are closely related to the relevance of an ACP and often used as an indicator.

3.3. Optimization of Access Control Policies

Building on the definition of ACP quality, we define ACP optimization as an improvement of the quality of existing ACPs with regard to specified quality criteria. This definition has two important implications: First, since ACP quality is multidimensional, ACPs cannot be optimized towards a universal optimum, but only towards a particular optimization objective. For example, consider an update operation that assigns a new permission to a role in order to provide this permission to the employees which inherit the role. This optimization may reduce the amount of missing UPAs and hence constitutes an optimization with regards to accuracy. At the same time, adding a new assignment to the role set increases its complexity and hence reduces its quality with regards to this quality dimension. Second, ACP optimization requires that an existing ACP set is updated rather than a new ACP set being created from scratch. This implication seems obvious, as the quality of a data state can only be improved in comparison with a reference state, i.e. the original data state that existed before the optimization. However, since every update operation generates a new data state, every optimization of an existing ACP set could also be interpreted as the creation of an new ACP set. The optimization of an existing ACP set differs in its goal from modeling a

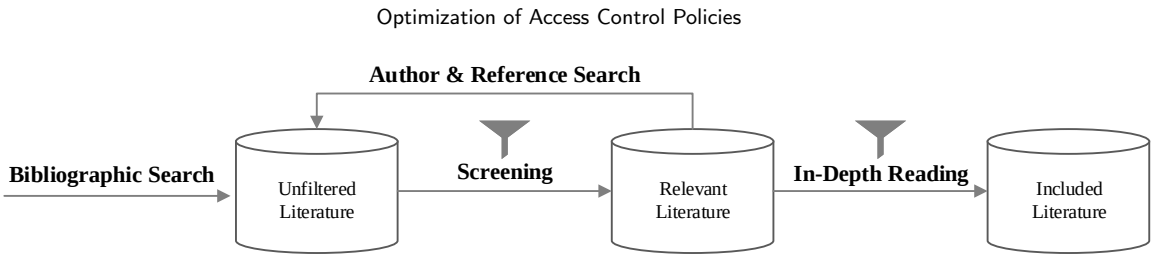


Figure 2: Applied Literature Research Process in accordance with Levy and Ellis (2006)

new one in that it aims to leave the existing set structurally intact. This goal stems from the need to maintain ACPs at a high quality level over time at a reasonable cost: Daily operations, such as an employee's department change, or the integration of new application systems into an organization's IT infrastructure, require frequent updates to an ACP set. Both neglecting such updates and sub optimal updates have a negative impact on the quality of the ACP set. Without continued quality measures, it is common for ACP sets to proliferate over time, while changing real-world conditions and proprietary policy updates cause their quality to degrade (Xia et al., 2014; Hu et al., 2011; Xu et al., 2017; Xiang et al., 2019). Unlike ACP modelling, ACP optimization aims to make partial changes to a (possibly very large) ACP set while leaving the remainder of the ACP set unchanged. As a result, an ACP set that was modelled with high quality can be kept on high quality with significantly lower effort than was required for its initial modelling. Moreover, by leaving an ACP set structurally intact, ACP optimization is able to retain the (often informal) semantic meaning of the existing ACPs that may be known only to human policy engineers. One possible approach to ensure that an ACP set remains structurally intact during optimization is the concept of "minimal perturbation", which is discussed in detail in section 5.3.

4. Literature Survey on Access Control Policy Optimization

4.1. Survey Methodology

The literature survey follows the methodology proposed by Levy and Ellis (2006). It aims to examine the existing body of research on ACP optimization (as defined in section 3.3) and provide a structured analysis of the research field. Since ACP optimization has to follow defined optimization objectives, we worked out six optimization objectives that are relevant and well established in existing literature. These six optimization criteria are presented along with further categorization criteria in section 4.2. We define the scope of literature included in the survey as *"Scientific publications that propose means for optimization of roles or attribute-based ACPs with regard to at least one of the six defined optimization objectives"*.

In accordance with the applied survey methodology, the literature research was started with a bibliographical database search. We used combinations of generic keywords like "maintenance", "improvement", "optimization", "correction" and more specific keywords for the distinct optimization objectives, such as "redundant" or "redundancy". Structured permutations of these keywords were entered in the online databases ACM Digital Library[4], IEEE Digital Library[5] and Google Scholar[6]. All publications that were not obviously related to another topic were added to an initial list of "unfiltered literature". We also included all publications from the ACM Symposium on Access Control Models and Technologies (SACMAT)[7] conference from the years 2001 to 2021 in this list. Every publication in the "unfiltered literature" list was then screened, which means that we read it superficially do determine whether it is relevant with regards to the survey topic. In this step, we read the title, abstract, introduction and conclusion and used the document search function to determine how the keywords applied during the bibliographical search were used. Doing so, we narrowed down the list of unfiltered literature to a second list of "relevant literature", which needs to address either the quality of ACP in general or one of the optimization objectives. All publications that were regarded relevant for the topic were then read in depth to determine whether they fit the survey scope and could be added the final list of "included literature". This list formed the ACP optimization literature catalogue which was categorized and analyzed. Since a bibliographical search could only serve as an entry point, we conducted author and reference search for all publications in the list of "relevant literature" and added the resulting publications back into the initial list of "unfiltered literature". Repeating the screening process for these publications, we executed a recursive search that allowed for a deep exploration of relevant research realms. A schematic overview of the applied literature research process is given in Figure 2.

Due to its broad scope, the survey comprised a heterogeneous literature base. As a result, the literature was hard to grasp with keywords. The majority of relevant results was yielded via author and reference search. Many publications in the "included literature" list do not explicitly define ACP optimization as maintenance or quality optimization, but rather define distinct objectives with a narrow scope which are semantically equivalent. Moreover, many relevant

[4] http://dl.acm.org/
[5] http://www.computer.org/
[6] http://scholar.google.com/
[7] http://www.sacmat.org/

Optimization of Access Control Policies

Table 2
Coverage of selected quality criteria in literature

Quality criterion	Coverage
Excessive UPAs	Fuchs et al. (2014); Hill (2006); Jaferian et al. (2014); Hummer et al. (2015); Puchta et al. (2019)
Missing UPAs	Benedetti and Mori (2018, 2019); Fuchs and Pernul (2010); Colantonio et al. (2012); Meier et al. (2013)
Redundancy	Guarnieri et al. (2013); Shamoon et al. (2012); Hu et al. (2013); Mitra et al. (2016); Kunz et al. (2015b)
Conflicts	Hounder (2010); Deng and Zhang (2017); Dia and Farkas (2012); Shamoon et al. (2012)
Complexity	Mitra et al. (2016); Kunz et al. (2015b); Servos and Osborn (2017); Currey et al. (2020); Fuchs et al. (2014); Molloy et al. (2010)
Grade of automation	Fuchs et al. (2014); Kunz et al. (2015a); Hu et al. (2010a); Kern and Walhorn (2005); Al-Kahtani and Sandhu (2004); Aftab et al. (2015)

research realms used other keywords than we would have expected: For example, the realm of "XACML anomaly analysis" aims at the identification (and sometimes removal) of anomalies in XACML policies, which can be either conflicts or redundancies and hence fit the survey scope. Since we did not know of this research realm beforehand, we hardly could have found it using keywords.

4.2. Research and Categorization Criteria

As seen in chapter 3, ACP optimization is not universal, but can only improve the quality with respect to defined optimization objectives. To define the research scope of the survey, we screened existing literature and selected six optimization objectives that are central for ACPs' fitness for use. These six optimization objectives serve as criteria for selecting relevant literature for this study (cmp. section 4.1) and are presented in detail in the following section 4.2.1.

Beside a textual analysis, the applied survey methodology suggests to categorize the literature catalogue. For a meaningful categorization, the selected criteria need to be concrete enough for in-depth insights while covering a heterogeneous literature catalogue. This leads us to three categories within this survey: (i) The first category is the targeted ACM of a publication. The majority of analyzed publications (except Hummer et al. (2015, 2016)) can be categorized as either RBAC or ABAC related. (ii) The next category is the optimization objective which serve simultaneously as research criteria. (iii) Another category is the contributed ACP optimization research artifact of the publication. (iv) Finally, the used data of the optimization method is analyzed as the last category. While the distinction between RBAC and ABAC is self-explanatory, the remaining categorization criteria are presented in sections 4.2.1 to 4.2.3.

4.2.1. Optimization objective

Existing literature defines several quality criteria on the basis of which ACPs can be optimized (cmp. section 3.2). To clearly define and narrow the scope of the survey, we searched the literature for optimization goals that are critical to the fitness for use of ACPs. Beckerle and Martucci (2013) conduct semi-structured expert interviews and a literature analysis to identify critical requirements for obtaining usable ACP sets. Based on their results, they argue that the main aim of ACP optimization should be to improve the accuracy and maintainability of ACPs. They specify these requirements and work out six optimization objectives that serve these two goals. The authors also develop metrics for the quantification of these criteria and conduct two user studies to evaluate them. To the best of our knowledge, this is the only scientific publication that documents a structured research process for developing ACP optimization objectives and provides a conclusive evaluation. The optimization criteria are: (i) "Allow no more than the owner wants to be allowed.", (ii) "Allow everything the owner wants to be allowed.", (iii) "A rule must not be fully covered by another rule of the same rule set.", (iv) "Two rules belonging to the same rule set must not conflict.", (v) "Minimize the number of rule set elements." and (vi) "Minimize maintenance effort in a changing system.". Each of these six optimization criteria can be mapped to one ACP quality dimension that was presented in section 3.2. We reformulate them to accord with the common literature terminology based on the addressed quality dimensions: *(i) Reduce excessive UPAs*, *(ii) Reduce missing UPAs*, *(iii) Reduce redundancy*, *(iv) Reduce conflicts*, *(v) Reduce complexity* and *(vi) Increase grade of automation*. We conducted a literature search to confirm the relevance of these six quality dimensions. Table 2 lists further publications that underline their importanc for ACPs' fitness for use. Note that this list is not exhaustive. Due to their thorough foundation in existing literature and their frequently argued importance for optimization, we selected these six optimization objectives as the basis for the literature survey.

4.2.2. Research Artifact

The analyzed publications on ACP optimization are heterogeneous not only in terms of the addressed optimization objectives, but also in terms of their contributions. We identified four types of research artifacts that are repeatedly presented to contribute to the optimization of ACPs: (i) Optimization process models analyze ACP optimization from a business perspective. They define process steps, roles and responsibilities and analyze how the technical optimization can be embedded into a changing real-world environment. (ii) Optimization algorithms define formal ways to modify

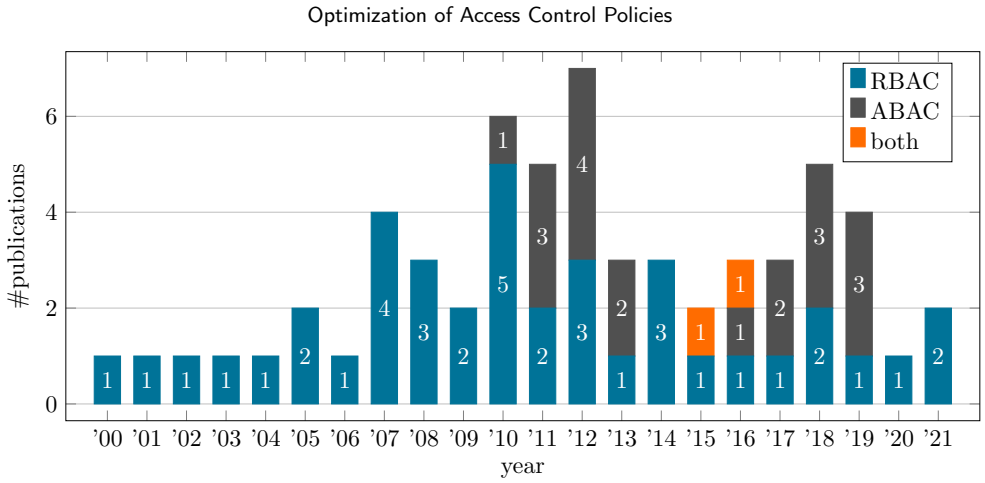


Figure 3: Publications over time

ACPs in order to improve their quality. They receive an existing ACP set (and possibly supplemental data) as input and generate an optimized ACP set as output. (iii) Optimization tools aid humans in the semi-automated optimization of ACPs. Publications with this kind of contribution present optimization tools to demonstrate how tool-supported ACP optimization can be done. (iv) ACM extensions propose ways to enhance ACMs to improve ACP quality. In the analyzed literature catalogue, this type of research artifact is only present in the form of RBAC extensions that enhance role definitions in order to provide automation. Note that we were not able to categorize the research artifact of a publication into one of these standardized categories in 5 cases.

4.2.3. Data usage

For the last criteria group, the survey literature was analyzed for the kind of data that was used to perform ACP optimization. We applied the conceptual IAM data model proposed by Kunz et al. (2019) to achieve an integrated view on the processed data. The model defines the central data entities that are processed in IAM and their relations towards one another. In addition to RBAC and ABAC, it integrates the conceptual entities defined in six central IAM technology standards[8] and proposes a terminology that is suited to cover the integrated concepts. According to this definition, a digital identity is a representation of a human user, which can be a record of an employee that is stored and processed in a human resource management system. Within an application system, digital identities possess accounts, to which permissions are assigned. Permissions can be hierarchically nested, in which case they are inherited transitively. Parallel to the definitions within an application system, digital identities can be assigned roles (as defined by RBAC), which inherit permissions through assignment and can also be hierarchically nested. Beside that, permissions can be granted via policies, which are ABAC or XACML policies in the context of this work. Note that in our terminology, both roles and policies are ACPs. The model defines a context entity, which is a scenario that can be evaluated by a policy (for example environmental conditions in an ABAC policy). At last, the model defines an attribute entity, which expresses a property of a digital identity, an account or a permission and can be evaluated by policies.

We identified a total of five types of data that were repeatedly used for ACP optimization: The *ACP set*, a *user-permission matrix*, *entity attributes*, *access logs* and *update logs*. From this set, we chose the use of entity attributes, access logs and update logs as categorization criteria. Processing of the ACP set and the user-permission matrix was not analyzed since these data types are trivial: The existing ACP set is the most basic type of data and must be known in order to be optimized. It comprises the roles or ABAC policies that are in effect and are updated in the course of the optimization. A user-permission matrix (also called access control matrix or UPA set) is the most basic representation of the permission assignments that an ACP set grants (Molloy et al., 2010). It is a boolean matrix which holds a true of false value for every possible user-permission combination. A user-permission matrix is commonly visualized in the form of an access grid and is not limited to a particular ACM (Meier et al., 2013). An optimization that updates an ACP set needs to verify that it did not create inaccuracies. Since it is the most basic permission assignment information available, a user-permission matrix is commonly assumed to be available for any ACP optimization effort.

Entity attributes are properties of entities other than the optimized ACPs themselves. As defined by Kunz et al, commonly expressed entities are digital identities, accounts and permissions. Attributes are often used to give ACPs semantic meaning: Since an attribute reflects a real-world property, binding an ACP to a given attribute can bind it to

[8]Lightweight Directory Access Protocol (LDAP), Security Assertion Markup Language (SAML)/Shibboleth, Service Provisioning Markup Language (SPML), Open Authorization (OAuth), System for Cross-domain Identity Management (SCIM) and XACML

its semantic meaning (Molloy et al., 2010). Optimization approaches that require entity attributes need to process actual value expressions of concrete entities. In contrast, a method that restructures a given XACML policy by reordering its attribute statements without processing any entity attribute values would not be classified into this category. The reliance on entity attributes is a limiting factor since their availability may be limited. Moreover, the use of entity attributes for ACP optimization means that the results are dependent on the value of said attributes at the time of the optimization, meaning that an optimization result may lose validity when the attribute values change.

Access logs express historic accesses of users to permissions. While their notations differ, access logs can be displayed as a tuple <*S,O,A,R*> that represents a historic access request, with *S* being the requesting subject, *O* being the requested object, *A* being the requested action and *R* the result of the request, i.e. *permit* or *deny* (Xiang et al., 2019). Note that some publications only consider successful permission invocations, thus reducing access logs to a tuple <*S,O,A*> of permitted access requests. Access logs provide valuable insight on the actual need of permissions and can help to identify missing or excessive permission assignments.

Update logs are the second type of historic data used for ACP optimization. Update logs contain information on past changes of IAM related entities, for example a modification of an ACP, the creation of a new user account or the change of an employee's department affiliation. Update logs can be used to identify real-world events that provide important ACP update information (for example, the job change of an employee might require a change of his or her permissions) and can provide insight on the development of an ACP set over time. To the best or our knowledge, no scientific publication exists that defines the structure of IAM update logs.

4.3. Criteria-Based Analysis

4.3.1. Overview

The literature survey yielded 61 publications that provide means for optimization of existing RBAC and ABAC ACPs since the year 2000. Out of these publications, 42 address the optimization of roles, 21 address the optimization of attribute-based policies, and two address both ACMs. The optimization of ACPs has been addressed continuously over the past 20 years. While the peak of interest for RBAC optimization occurred between 2007 and 2014, the topic receives steady attention to this day, underlining that role optimization remains a relevant research subject. The first analyzed ABAC optimization paper was published in 2010. The tables 3 and 4 present all analyzed publications and their categorization according to the survey criteria defined in section 4.2. Figure 3 depicts all analyzed sources ordered by their year of publication.

4.3.2. Considered Optimization Scenarios

To connect ACP optimization with real-world scenarios, it is helpful to consider optimization scenarios. While various themes for optimization scenarios are possible, e.g. IAM goals (Hummer et al., 2018) or usage of popular techniques like access reviews (Groll et al., 2021), it is reasonable to take a closer look at maturity and automation as its driver. Like shown by Schrimpf et al. (2021) a driver for higher maturity is automation which requires or builds upon underlying optimization methods. In the sense of automation, ACP optimization scenarios can thus be considered as manual, semi-automated and automated.

In manual optimization, a human administrator or policy engineer updates part of an ACP set based in their individual context knowledge. Manual adjustments to an ACP set are commonly done in daily operations, e.g. to grant employees new permissions after their responsibilities changed. Another example are access reviews, where a responsible human (e.g. a department head) tries to find excessively assigned permissions manually and marks them for revocation (Jaferian et al., 2014). Due to the lack of automation, manual optimization is limited to small ACP sets or subsets of larger ACP sets. Another optimization scenario is the fully automated updating of an ACP set. In this scenario, an ACP optimization algorithm generates optimization steps for an ACP set. The resulting changes will be implemented in the underlying applications automatically. This approach has the disadvantage that an algorithm has no understanding of the semantic structure of an ACP set. As a result, the structure of the optimized ACPs may differ greatly from their original structure. In addition, fully automated methods can hardly be integrated into established change management processes, which require that changes to ACPs must be confirmed by a responsible employee. The third scenario is semi-automated optimization. Semi-automated processes try to bridge the gap between manual and automatic optimization by enabling humans with technical support to optimize a very large set of ACPs. A common example are recommendation-based optimization methods, which generate possible ACP updates automatically, and delegate them to responsible humans for decision. Updates will only become effective if the responsible human agrees to them. Semi-automated optimization can be supplemented by ACP visualization procedures and other data analysis techniques which are not optimization methods themselves. The concept of recommendation-based optimization is discussed in section 5.4.

The publications analyzed during the survey cannot always be clearly assigned to one of these scenarios. We observe that publications that propose an optimization algorithm typically assume fully automated optimization. Still, some optimization algorithms allow that the resulting change steps are delegated to humans in the form of recommendations, thus enabling them for semi-automated optimization (Han et al., 2012; Hu et al., 2010a; Rao et al., 2021; Benedetti and Mori, 2018, 2019). Publications that present an optimization tool aim to support humans in the semi-automated optimization of ACPs by definition. Of

Optimization of Access Control Policies

Table 3
Categorized literature for ABAC optimization

Publication	Optimization Objective						Research Artifact				Data Usage		
	Reduce Excessive UPAs	Reduce Missing UPAs	Reduce Complexity	Reduce Redundancy	Reduce Conflicts	Increase Grade of Automation	Optimization Process Model	Optimization Algorithm	Optimization Tool	ACM Extension	Entity Attributes	Access Logs	Update Logs
Argento et al. (2018)	■							■			■	■	
Benkaouz et al. (2016)			■					■					
Cheng et al. (2019)					■								
Deng and Zhang (2017)					■			■					
Dia and Farkas (2012)					■			■					
Guarnieri et al. (2013)				■				■					
Hadj et al. (2017)			■	■				■					
Hadj et al. (2018a)				■	■			■				■	
Hadj et al. (2018b)				■	■			■					
Hadj et al. (2019)					■			■				■	
Hein et al. (2011)					■			■					■
Hounder (2010)					■			■					
Hu et al. (2011)				■									
Hu et al. (2012)				■	■			■					
Hu et al. (2013)				■	■			■					
Hummer et al. (2015)	■	■					■				■	■	
Hummer et al. (2016)	■	■					■				■	■	
Narouei and Takabi (2019)			■					■			■		
Oberholzer (2011)			■		■			■					
Shamoon et al. (2012)					■			■					
Stepien et al. (2012)			■	■				■					

the publications analyzed that present a process model, all except Strembeck (2010) assume that changes to an ACP set can be generated by algorithms, but must be delegated to a human for decision before becoming effective. It is noticeable that publications that aim to embed ACP optimization into an organization's processes largely reject fully automated optimization (except Strembeck (2010)). Publications that propose an extension of an ACM define ways to extend the structure of a role set with automation logic. However, they do not define a concrete optimization scenario in which a role set is updated to inherit this automation logic. ACM extensions are thus rather a blueprint for ACP optimization and not limited to a concrete optimization scenario.

4.3.3. Reduce Excessive & Missing UPAs

We identified nine publications that aim at identifying and rectifying inaccurate UPAs. Out of these nine publications, six address both missing and excessive permission assignments. Another two exclusively aim at excessive permission assignments and one addresses only missing permission assignments. Fuchs et al. (2014) propose a process model for RBAC optimization. Hummer et al. (2015, 2016) present a process model for both RBAC and ABAC optimization. Benedetti and Mori (2018) present both an RBAC optimization process model and a Max-SAT algorithm that uses access logs to identify missing permission assignments and adjust roles while minimizing their complexity. They expand it in a follow-up publication to address excessive permissions as well (Benedetti and Mori, 2019). Baumgrass (2011) and Zhang et al. (2013) both use access logs to identify missing or excessive UPAs and adjust roles accordingly. Argento et al. (2018) use access logs to identify excessive permission assignments and update ABAC policies. Groll et al. (2021) propose to use negative access review decisions, i.e. decisions in which a human reviewer identified excessive UPAs, to identify similar UPAs and generate revocation recommendations for them, thus amplifying the impact of manual identification of excessive permissions. To the best of our knowledge, this is the only publication proposing a method that uses a data source other than access logs to identify excessive permissions automatically. Note that approaches which provide automation for RBAC updating are also relevant for this optimization objective as they aid the closely related objective of *preventing* UPA inaccuracies. These approaches are analyzed in section 4.3.6.

The key challenge for correcting missing or excessive UPAs is to identify them. Once found, such inaccuracies can be corrected in a fully automated manner. Throughout the literature analysis we identified three basic approaches for identifying missing or excessive permission assignments: First, manual identification requires that a human overlooks

Optimization of Access Control Policies

Table 4
Categorized literature for RBAC optimization

Publication	Optimization Objective						Research Artifact				Data Usage		
	Reduce Excessive UPAs	Reduce Missing UPAs	Reduce Complexity	Reduce Redundancy	Reduce Conflicts	Increase Grade of Automation	Optimization Process Model	Optimization Algorithm	Optimization Tool	ACM Extension	Entity Attributes	Access Logs	Update Logs
Aftab et al. (2015)						■				■	■		
Al-Kahtani and Sandhu (2002)						■				■	■		
Al-Kahtani and Sandhu (2003)						■				■	■		
Baumgrass (2011)	■	■										■	
Benedetti and Mori (2018)		■	■				■	■				■	
Benedetti and Mori (2019)	■	■	■				■	■				■	
Chakraborty and Ray (2006)						■				■	■		
Fuchs et al. (2014)	■	■	■				■					■	■
Gal-Oz et al. (2011)				■				■					
Groll et al. (2021)	■										■		■
Guo et al. (2008)			■					■					
Han et al. (2012)						■		■		■	■		
Herzberg et al. (2000)						■				■	■		
Hu et al. (2010a)						■	■	■	■				
Hu et al. (2010b)						■		■					
Hu et al. (2016)						■		■					
Huang et al. (2012)						■		■		■	■		
Hummer et al. (2015)	■	■					■				■	■	
Hummer et al. (2016)	■	■					■				■	■	
Kern and Walhorn (2005)						■				■	■		
Lu et al. (2014)						■							
Lu et al. (2017)						■				■			
Molloy et al. (2010)			■	■				■			■		
Ni et al. (2009)						■				■	■		
Pan et al. (2018)			■					■				■	
Pang et al. (2007)				■	■			■					
Pang et al. (2008)			■	■				■					
Parkinson et al. (2020)						■				■			
Rao et al. (2021)						■		■		■		■	
Saffarian et al. (2009)						■				■	■		
Shafiq et al. (2012)					■			■					
Sheng and Osborn (2004)						■				■	■		
Strembeck (2005)				■					■				
Strembeck (2010)				■			■					■	■
Takabi et al. (2007)						■		■		■	■		
Takabi and Joshi (2010)			■					■					
Vaidya et al. (2008)						■		■					
Xia et al. (2014)			■					■					
Yi-qun et al. (2007)						■				■	■		
Zhang et al. (2007)			■					■					
Zhang et al. (2013)	■	■						■				■	
Zong et al. (2011)						■				■	■		

an ACP set and tries to find assignments which he or she knows to be inaccurate. This task is commonly executed in the form of access reviews, which are the de-facto standard process for this task and mandated by central regulation frameworks. Second, usage-based approaches analyze access logs for information on historic permission invocations and can thus be used to determine the set of actually needed UPAs. Third, update history based approaches analyze historic updates of the ACP set to identify missing or excessive permission assignments. The advantages and data requirements of these basic concepts are discussed in section 5.1.

4.3.4. Reduce Complexity

The survey yielded ten publications that aim to reduce the complexity of an RBAC state and five publications

that aim to simplify an ABAC state. Out of the RBAC related publications, Fuchs et al. (2014) are the only one that addresses the problem exclusively at the process level. They propose a process model for the optimization of roles regarding accuracy and complexity. Benedetti and Mori (2018, 2019) define an RBAC maintenance process model and an algorithm that optimizes the complexity of an RBAC state. Xia et al. (2014) define the Role Refinement Problem and provide an algorithmic analysis, but also define process-level requirements for role refinement. The remaining six RBAC-related publications address complexity optimization strictly on a technical or algorithmic perspective. Takabi and Joshi (2010), Pan et al. (2018) and Benedetti and Mori (2018, 2019) aim to minimize the amount of updates required to perform the generated optimizations. Molloy et al. (2010) use user attributes to increase the semantic meaning of the roles that are either mined or maintained in their approach. They also argue for the use of access logs and update logs, but do not use them themselves. Pan et al. (2018) use access logs to calculate the usage of roles, which they use to determine its relevance as one indicator of its quality. The five ABAC-related publications propose algorithms to reduce the complexity (Oberholzer, 2011; Benkaouz et al., 2016; Narouei and Takabi, 2019) or both complexity and redundancy (Stepien et al., 2012; Hadj et al., 2017) of an existing ABAC state. Hadj et al. (2017) and Narouei and Takabi (2019) calculate the similarity of input and output state to minimize the required updates. Narouei and Takabi (2019) require user and permission attributes for the refinement of an ABAC policy set. Altogether, ACP optimization for complexity requires no data other than the ACP set and the user-permission matrix since it is a mere algorithmic problem that can be solved without semantic knowledge. As long as the resulting UPAs remain unchanged, the ACP set can be rearranged for a lower complexity in a fully automated manner. However, context knowledge (for example provided by access logs) can provide further inside on the meaning of ACP and benefit the optimization efforts.

4.3.5. Reduce Redundancy

We identified 14 publications that propose methods for reducing redundancy in existing roles or attribute-based policies. Strembeck (2005) define a high-level maintenance process for roles that aims at reducing redundancy. Strembeck (2010) present a role engineering tool that supports humans in finding and removing redundant role-permission assignments. The remaining twelve publications propose algorithms for the reduction of redundancy in roles or XACML policies. We observe that optimization methodologies often consider redundancy reduction a secondary optimization, as four of the analyzed publications aim to reduce complexity and five rectify conflicts as their primary objective. While the reduction of redundancy is a non-trivial algorithmic problem, it can be addressed very well with algorithms since solving it at core requires no data other than the optimized ACP set and its UPA set. Consequently, most of the analyzed approaches use no further data. However, Molloy et al. (2010) use entity attributes to mine new and optimize existing roles with redundancy as one optimization objective. Strembeck (2005, 2010) propose the use of context constraints during role optimization and define a process for engineering it, using access logs and update logs as data sources.

4.3.6. Reduce Conflicts

A large research area aims at the resolution of ACP conflicts. However, most of these publications propose methods to generate a permit or deny decision during the evaluation of ACPs despite the existence of a conflict. They do not modify the underlying ACPs to correct the error and hence to not fit into the scope of this survey. Altogether, we identified 14 publications that propose methods for the reduction of ACP conflicts. Cheng et al. (2019) define a methodology for removing inconsistencies (which include conflicts) in rule-based ACPs such as ABAC policies. Despite their method being named "removing process", we chose not to categorize it as a process model as it is a technical methodology that is closer to defining an algorithm than a (business) process model. The remaining 13 publications define conflict reduction algorithms. Both Pang et al. (2007) and Shafiq et al. (2012) propose graph optimization algorithms to remove constraint conflicts on an RBAC state. The remaining eleven publications propose algorithms for the reduction of rule conflicts in an XACML policy set.

A crucial challenge with direct impact to the ACP quality is to decide whether a given conflict should be converted into a permit or deny decision before updating the ACP set accordingly. A wrong correction decision would still remove a conflict in the ACP set, but lead to a UPA inaccuracy in return. The analyzed literature offers different approaches to make this decision. Hu et al. (2011, 2012) and Deng and Zhang (2017) embed existing conflict resolution strategies in their approach. Hu et al. (2013) refine the previously proposed approach to apply conflict resolution strategies for individual segments of an ACP set, and apply removability constraints to individual ACPs in order to achieve a more fine-grained correction decision. Both Dia and Farkas (2012) and Hadj et al. (2018b) extend ACPs with scope constraints to support the correction decision. Dia and Farkas (2012) also generate recommendations for decisions on conflict removal which can be delegated to responsible entity owners if the correction algorithm generated a non-mandatory update. Hein et al. (2011) generate update logs that track the changes to an ACP and use these in order to support administrators and allow them to restore a previous state which is known to be correct. Shafiq et al. (2012) apply priorities to role constraints and evaluate these during the conflict correction. At last, Hounder (2010) defines algorithms to aid human policy administrators in conflict correction, thus proposing a semi-automation of conflict correction which can incorporate the semantic knowledge of a human in the resolution decision.

We observed that conflict correction research primarily focuses on attribute-based policies with only two publications for role conflict correction yielded by the survey. We

explain this by the fact that attribute-based ACPs are prone to conflicts since they define both positive and negative authorizations by default. Roles themselves in contrast define only positive authorizations and can only be conflicted if used with constraints (Ahn and Sandhu, 2000). Nevertheless, role constraints are crucial part of role-based access control and are required to express role restrictions for fundamental process-level requirements such SoD policies. The current state of research on RBAC conflict correction does not reflect this.

4.3.7. Increase Grade of Automation

Since ABAC is dynamic by nature, providing automation is only relevant for RBAC in the scope of the survey. Out of 22 publications that provide means for RBAC automation, we group 16 publications into three classes: Rule-based automation, learning-based automation and trust-based automation.

Rule-based automation extends roles with rules which evaluate attributes to automatically update employee-role assignments (Al-Kahtani and Sandhu, 2002; Kern and Walhorn, 2005; Yi-qun et al., 2007; Han et al., 2012; Huang et al., 2012; Aftab et al., 2015), role-permission assignments (Han et al., 2012; Huang et al., 2012; Aftab et al., 2015; Parkinson et al., 2020) or role hierarchy assignments (Al-Kahtani and Sandhu, 2003). They can hence be interpreted ad hybrids of RBAC and ABAC. Since attributes can reflect a real-world property with semantic meaning, attribute-based role automation can help to improve the semantic meaningfulness of roles. A single rule can cover a multitude of role model updates and can hence greatly reduce administrative effort for a role set. Nevertheless, automation rules can outdate over time and need to be maintained together with the role set.

Learning-based automation approaches use existing entitlement information as input to find valid role updates. Sheng and Osborn (2004) use entity attributes to generate employee-role updates and Ni et al. (2009) use entity attributes to generate role-permission updates. Rao et al. (2021) utilize access logs to automatize employee role assignment. Learning-based automation works without static rules and hence does not necessarily require manual definitions. Unlike rule-based automation, learning-based automation may react to changes that the role engineers did not consider when modelling the role set. However, learning-based automation requires learning data as input, and the quality of its updates is limited by the quality of the input data. As a result, we argue that learning-based optimization might be better suited to maintain a role set with an already high quality, than to optimize a role set with low quality from scratch.

Trust-based automation assigns roles to users based on the users' trustworthiness (Herzberg et al., 2000; Zong et al., 2011; Chakraborty and Ray, 2006; Takabi et al., 2007; Saffarian et al., 2009). Trust-based approaches differ from the two previously presented ones as they do not assign roles based on the tasks that a user has to perform (as required by the principle of least privilege), but try to assess a set of maximum permissible permissions. This is typically done by calculating trust scores for users and defining minimum trust scores that a user needs to have in order to be granted certain roles or permissions. Trust-based automation approaches are designed for open environments where the users are not completely known (for example collaborative platforms like Wikis). They are not designed for classical inhouse identity management environments where employees with a defined task range are managed (cmp. Fuchs and Pernul (2007); Fuchs et al. (2009)), but aim to mitigate risk when little user information is available.

We identified another six publications that provide means for automatic role updating, but do not try to identify possible updates themselves: Hu et al. (2010b) generate migration paths to automatize the implementation of role model updates in application systems. Hu et al. (2010a) define a tool and a process which aid the automatic updating of role-permission assignments by checking whether an update is achievable with a given set of constraints. Lu et al. (2014) and Hu et al. (2016) evaluate role updating algorithmically to determine the complexity of automatic checking for role-permission and role-role assignments. Lu et al. (2017) propose a role generalization algorithm that aims to optimize roles for automatic assignment via user authentication queries. Vaidya et al. (2008) aim to enable role mining algorithms for optimization of an existing role set by generating a state that is as similar as possible to an existing role set and an optimal one.

Altogether, we conclude that RBAC automation is thoroughly covered by research. It can help to reduce administrative effort and maintain a high role model quality. However, it does not make role maintenance obsolete, since all types of automation have limitations and cannot be expected to react to all future changes adequately. Beside a limited scope, RBAC automation configurations themselves can be erroneous or outdate over time just like a classical role set.

5. Discussion

Building on the results of the literature survey, this chapter discusses important aspects of ACP optimization. In doing so, we analyze several concepts that are repeatedly addressed in the survey literature and play a critical role in the optimization of ACPs. Since the identification of UPA inaccuracies is the most critical challenge in correcting excessive and missing UPAs, we work out three prototypical approaches commonly found in the literature and discuss their advantages and limitations as well as their data requirements in section 5.1. We then examine the availability of the three types of data that were included in the literature analysis in section 5.2. Subsequently we discuss the concepts of minimal perturbation and of recommendation-based optimization in sections 5.3 and 5.4. At last the limitations of this work are discussed in section 5.5.

5.1. Identification of UPA Inaccuracies

Finding inaccurate UPAs is the key challenge when optimizing ACPs for accuracy. If the complete set of required UPAs is known, an ACP set can be optimized for perfect accuracy in a fully automated manner. However, such a set does not usually exist in an explicit form, and finding out which permissions any subject should be granted in accordance with an organization's security policy is difficult and time-consuming. ACP accuracy optimization methods hence face the primary challenge to identify as many inaccuracies as possible in order to be able to correct them. During the literature analysis we identified three prototypical approaches to identify UPA inaccuracies, which we present and discuss below.

5.1.1. Manual Identification

Manual identification primarily concerns excessive UPAs, because manual identification of missing UPAs can simply be initiated by the affected users, for example by ordering a missing permission in a structured process through an IAM entitlement shop (Hornsteiner et al., 2020). The most obvious approach to identify excessive UPAs is to have a human who knows an organization's security policy check the effective UPAs and search for inaccuracies. A process that works by this scheme is known by the name of *Access Reviews*. Access Reviews are a standard IAM process that is executed periodically and aims to identify and rectify excessive UPAs (and sometimes other data inaccuracies like inaccurate attribute values) (Jaferian et al., 2014). Their execution is strongly driven by external requirements raised by compliance frameworks or IT security standards (Fuchs and Pernul, 2007; Royer, 2008). During an Access Review, a responsible human reviews a list of entitlement assignments for the users, ACPs or permissions within their responsibility, and decides whether they are still necessary. Based on this decision, the reviewer will either confirm an assignment's correctness, or refuse to confirm it, in which case the assigned permissions will be revoked. Access Reviews are a central measure to prevent the accumulation of excessive permissions (Jaferian et al., 2014). However, they are error-prone and tend to overlook and confirm excessive UPAs due to a number of structural challenges (Groll et al., 2021). In particular, the sheer amount of data that has to be processed during Access Reviews can be overwhelming, and the decision whether a user requires a particular permission is difficult to make. In case of uncertainty, Access Review decisions are biased towards confirming an existing assignment for a number of reasons: (i) Since an existing UPA likely has been subject to an approval process before becoming effective, a reviewer has reason to believe that an existing assignment has a legitimate reason. (ii) If a reviewer makes a false decision, only a false revocation of an existing UPA would have an immediate consequence, because an employee would no longer be able to execute a certain task as a result. In contrast, an erroneous confirmation of an already granted UPA is unlikely to have an immediate effect as long as the resulting security vulnerability is not abused for malicious action. (iii) For the same reason, the visibility of an erroneous revocation within an organization is higher than the visibility of an erroneous confirmation, which adds a social incentive for reviewers to simply confirm existing UPAs as to avoid visible errors. As a result, the effectivity of Access Reviews is limited. While some research effort tries to aid users in the effective execution of Access Reviews (Jaferian et al., 2014; Bobba et al., 2005), to the best of our knowledge only one approach was proposed that aims to measure the quality of Access Review decisions in order to identify erroneous UPA confirmations automatically (Groll et al., 2021). Despite the importance of Access Reviews and the difficulties practitioners face in implementing them, the tasks of aiding companies and users in the execution of Access Reviews and measuring their effectively were scarcely addressed by research.

5.1.2. Usage Based Identification

Usage based identification of UPA inaccuracies evaluates historic permission invocations to determine which permissions a particular user actually needed in the past. It is hence directly related to the principle of least privilege, which states that any user should not inherit more permission than necessary to perform his or her tasks. Several ACP optimization approaches use access logs for usage based identification of UPA inaccuracies (cmp. section 4.3.3). If access logs are available, excessively assigned permissions can be found relatively easy by comparing historic permission usages with the actually granted permissions of an ACP set. Any permissions that are granted for a user, but were not used within a specified time frame (e.g. over the last year) can be interpreted as excessively assigned and are hence candidates for removal. This approach is well suited to detect large quantities of excessively assigned permissions that were overlooked during UPA maintenance. However, it can struggle to distinguish excessive UPAs from accurate, but rarely used ones (e.g. a yearly creation of a report). Moreover, this approach can only identify excessive UPAs as long as they are not used (possibly even with malicious intent).

If access logs do not only include historic permission invocations, but also access requests which have been denied, they can be used to identify missing UPAs: If a permission invocation is often requested and denied, this can be an indicator that the permission is handled too restrictively. However, the identification of missing UPAs leaves more room for interpretation than the identification of excessive UPAs and requires finding a balance between business continuity considerations and security considerations. We hence argue that the paramount value of access logs lies in identifying excessive UPAs. Especially UPA accumulation, i.e. the accumulation of permission assignments that were once granted but are no longer required by a user (for example because the user's responsibilities have changed since the permission assignment) is a major problem that can be addressed very well with the described approach. However,

usage based correction is limited by the availability of access logs, which is discussed in section 5.2.

5.1.3. Update History Based Identification

The use of update histories for the optimization of ACP accuracy was proposed by several authors. Some also provided examples how update histories could be used for this cause. Fuchs et al. (2014) cite the update history of a role model as an important source of context information for optimizing a role model. However, they only consider it for manual evaluation by human role engineers. Strembeck (2010) names trace management as necessary requirement for maintaining complex models, but does not go into the details of ACP maintenance in this regard. Hein et al. (2011) propose an algorithm to generate update logs that track the changes made to an ACP set and use them to enable administrators to perform a rollback operation and restore previous states. Mitra et al. (2016) propose to use update logs to identify roles which decayed over time in order to select candidates for maintenance. While the authors do not elaborate on this, using historical data to assess timeliness is common practice in other areas of data quality research (Heinrich and Klier, 2009). Molloy et al. (2010) name the event pattern of a user losing several permissions, followed by being assigned several permissions within a short time frame as an indicator for a job change event and argue that such events provide valuable context information for the creation of high-quality roles. While not explicitly naming it as a role optimization use case, they also stress that the creation of high-quality roles is not a once-and-for-all effort, but must be succeeded by continuous role optimization efforts. They provide further examples for meaningful events within a role model's update history, arguing that permissions which are often assigned or unassigned together are likely related to the same real-world context, thus providing an example of using update logs to determine the semantic meaning of permissions. Molloy et al. (2010) also argue that historic update information provides evidence on legacy permissions which should be removed from the role model. To the best of our knowledge, no methods were proposed that use update logs for the correction of missing or excessive UPAs. However, Groll et al. (2021) propose an approach to find erroneous Access Review decisions (i.e. decisions where an excessive permission assignment was falsely confirmed) automatically. The proposed approach, which uses revocation decisions from the analyzed Access Review as input, performs learning-based outlier detection to find other Access Review decisions which are likely to be over-permissive. Although the method does not explicitly use historic update information, it can be abstracted as an analysis of UPA unassignment events, which could also be taken from update logs instead of Access Review decisions. Altogether we conclude that ACP update histories are a promising source of context information for identifying excessive or missing UPAs. To this day, existing research has made few attempts to make use of this information source.

5.2. Data Availability

To highlight the limitations resulting from the data requirements of the analyzed ACP optimization approaches, we will discuss the availability of the investigated data types. As defined in in section 4.2 we assume the conceptual IAM model of Kunz et al. (2019) for an integrated view on the processed data. The three classes of data for which we analyzed the survey literature catalogue are entity attributes, access logs and update logs. First of all, the use of all of these data sources requires an integrated view on the processed IAM data. This requirement is not trivial since an identity management infrastructure comprises a wide range of heterogeneous data sources. These data sources may either provide a centralized data view (e.g. a human resource system that stores the employee data for the entire organization or a directory system that serves as a centralized user account and permission data storage) or exist as numerous decentralized data sources that have different data schemes and data storages (e.g. application systems that manage individual user accounts and permissions for their own application context) (Fuchs and Pernul, 2007). While having an integrated data view on the identity management infrastructure is a common prerequisite for IAM measures, the creation of such a view requires significant effort and should therefore not be neglected (Fuchs et al., 2009).

5.2.1. Availability of Attributes

Attributes are properties of entities within the IAM data view other than the ACPs themselves. If an integrated view of the related entities exists, entity attributes are from a technical point of view easily obtainable. Due to the sensitivity of personal data however, the processing of attributes of digital identities may be restricted. While this means that sensitive attributes (like an employee's loan details or sick leave history) might not be available for ACP optimization, it should not prevent attributes of digital identities from being processed altogether, as many central business-related attributes (such as departmental affiliation and job title) do not fall into this category.

5.2.2. Availability of Access Logs

The availability of access logs is more complex. The OASIS XACML reference architecture presumes that authorization requests are sent to a central access control mechanism with a Policy Decision Point (PDP) at its core. If this is the case, access requests (both granted and denied ones) can be logged completely. However, we argue that this is a prototypical architecture which comes with several challenges: (i) Creating a central access control mechanism requires that all applications which are subject to IAM efforts delegate their authorization decisions to a PDP. This requires significant integration effort, which may not be economically feasible. (ii) The central evaluation of access requests requires, that every authorizable action of a user within an application (which may in case of highly configurable systems come down to every single click on a button) be sent to the PDP, evaluated and answered in real-time before the user action is

executed. Since this has to happen during the run time of the application with negligible time delay, this requires a highly traffic resistant, performant and available IT infrastructure and almost immediate access request evaluation on the PDP side, again implying efforts that may very well exceed the benefits of centralizing authorization decisions. (iii) This approach requires that all applications which are subject to IAM are technically able to delegate every authorization decision to a PDP, which is currently not the standard.

The alternative approach to obtain access logs is by decentralized logging within the applications where the permission invocations occur. The decentralized logs must then be collected and integrated in a central log stream. While the decentralized approach does not require real-time decision delegation, it also requires high integration effort and is limited by the ability of all managed applications to log permission invocations: Although several industry software solutions do provide access logs or access statistics[9] , this is not a primary use case for many software products and hence not the standard.

Beside technical and economical limitations, access logs contain particularly sensitive personal data as they enable work monitoring, and their use may be restricted due to requirements of the legislator or employee representatives. This problem is also known in other fields which monitor IT infrastructure for security, for example in the context of Security Information and Event Management (SIEM) (Menges et al., 2021). Due to the sensitivity of the processed data, it may be necessary to limit surveillance to the most critical areas, for example, users with a particularly high number of privileges (like administrators) or critical applications (like a banking system). Overall, we conclude that access logs are difficult to develop as a data source. In addition to significant technical hurdles, the economic viability of monitoring activities across the board is not always given. Moreover, ethical and legal hurdles must be considered. For these reasons, we argue that the availability of access logs that include all user accounts and permissions from all the applications in the scope of an organization's IAM cannot be assumed to be standard. Approaches that rely on the availability of access logs for optimization of ACPs may have limited applicability in practice.

5.2.3. Availability of Update Logs

Update Logs contain the update history of the entities processed in IAM. All availability limitations that apply to entity attributes hence also apply to entity attributes within the update logs. Apart from that, update logs are easy to obtain: If an integrated view of the IAM data exists, then its changes can be monitored, too. The creation of update logs is a common functionality for industrial IAM systems, which have a central view over IAM entities since they are used to manage ACPs and to provision them to the related application systems. Moreover, legal regulations imply the (partial) existence of update logs: In order to check compliance with the principle of least privilege, an auditor must be able to investigate when and how an ACP set was updated to grant users new permissions. For this reason, we argue that update logs are a readily accessible data source. Nevertheless, to the best of our knowledge, no scientific model exists that describes how IAM update logs are structured. Foundation work is still missing for the scientific development of this data source and possible applications.

[9]For example, Microsoft Exchange (https://www.microsoft.com/de-de/microsoft-365/exchange/email) provides detailed logs on email distribution list usages and SAP ERP (https://www.sap.com/) provides aggregated statistics of transaction invocations.

5.3. Minimal Perturbation

The concept of minimal perturbation aims to optimize ACPs with as few changes as possible (Vaidya et al., 2008). ACP optimization algorithms often address this goal by defining maximal similarity to the original state as a secondary optimization objective. The ability to optimize ACPs with few changes is an important factor that can determine the practicality of an optimization method for two reasons: (i) The fewer changes needed to achieve quality improvement, the lower the administrative effort required to implement those changes. Since many organizations are mandated to execute approval processes for changes to the ACP set, an ACP update often requires the interaction of humans (e.g. a role owner or employee owner) before it can be enforced. Moreover, since access control is often enforced decentralized within the application systems managed in an identity management infrastructure, changes to the ACP set have to be provisioned into the related application systems in order to take effect. Minimizing the perturbation hence improves the economic viability of ACP optimization. (ii) The fewer changes needed to achieve quality improvement, the higher the degree to which an ACP set remains structurally intact. ACPs are typically modeled to reflect semantic concepts. This goes from single policies reflecting simple statements (like "any employee of this organization may access the WiFi hot-spot") up the whole ACP set, which can be designed, for example, to reflect the organizational structure of a company (Fuchs and Pernul, 2008; Xu, 2014). Furthermore, ACPs incorporate (often informal) contextual knowledge that may only be known to human policy engineers. By largely preserving the structure of an ACP set after its initial creation, the semantic meaning of the ACPs and the contextual knowledge that has gone into them are also preserved, which is beneficial for the quality of the resulting ACP set and prevents it from becoming unrecognizable after repeated execution of optimization methods. Since minimal perturbation can be critical for the practical viability of an optimization method, we argue that it is not adequately addressed by existing research: Out of 38 publications in the literature survey catalogue that propose an algorithm for ACP optimization, only 10 consider minimal perturbation when defining their optimization objectives (Benedetti and Mori, 2018, 2019; Hadj et al., 2017; Hu et al., 2010a; Pan et al., 2018; Narouei and Takabi, 2019; Rao et al., 2021; Takabi and Joshi, 2010; Vaidya et al., 2008; Zhang et al., 2013).

5.4. Recommendation-Based Optimization

In this section, we discuss the concept of recommendation-based optimization. Recommendation-based optimization means that ACP update steps produced by an optimization method are not applied directly to the underlying ACP set, but bundled into optimization recommendations. An optimization recommendation represents one or more changes to the ACP set and can be delegated for decision to responsible human decision makers. Only when the decision maker agrees to the optimization, the associated update steps are applied to the ACPs set. If no manual decision is required, an optimization recommendation can alternatively also be executed automatically.

The use of recommendations addresses business-related constraints that can be crucial for the practical applicability of an optimization method: (i) Approval processes often require that changes to the policy set be approved by a responsible subject (like a department head, an application owner or policy owner) before being implemented. By bundling optimization decisions into human-decidable steps, organizations can use (semi-)automatized optimization methods while remaining compliant. (ii) Business constraints may exist (like regulatory requirements or practical hurdles) that prohibit some changes in the policy set, but are unknown to optimization algorithms. Moreover, automatic optimization methods struggle to understand the semantic structure and context meaning of an ACP set. As a result, updates may be proposed that are technically valid, but make little sense in the real world. With recommendations, a human decider can serve as a quality gate for optimization steps and prevent changes that are problematic or forbidden. Since domain experts are likely to have specific context knowledge, including them into the optimization process can improve the effectiveness of the optimization altogether. We argue that recommendation-based ACP optimization constitutes a hybrid model of fully-automated and manual optimization. In the closely related domains of RBAC and ABAC policy modelling, which face the closely related challenge to automate the creation of semantically meaningful ACPs with high quality, hybrid approaches have also been proposed and gained broad acceptance (Fuchs and Pernul, 2008; Das et al., 2018).

A number of optimization methods analyzed in the survey rely on recommendations. Fuchs et al. (2014) propose a process model for RBAC optimization that includes a mechanism for generating new role extensions (i.e. new assignments of roles to employees, permissions or other roles). They define that every role extension is delegated to a role owner for decision. Hummer et al. (2015, 2016) propose a process model for the optimization of RBAC or ABAC ACPs based on usage patterns. Similar to Fuchs et al., they define that every optimization step is processed via a recommendation mechanism. Benedetti and Mori (2018) define an RBAC maintenance process that identifies missing role-permission assignments algorithmically. In Benedetti and Mori (2019), this process is extended to also find excessive permission assignments. They stress that the found "violations" must be presented to a security administrator who may confirm or reject them before they are processed to generate RBAC model updates. This approach differs from the previous recommendation-based optimization approaches as the recommendation mechanism is active in a preliminary stage of the optimization process, not at the end of it. Groll et al. (2021) define a methodology that analyzes confirmation decisions of access reviews to find possible errors. The results are then delegated to a human policy analyst for inspection. If the analyst confirms the error, the underlying UPA is revoked, which means that every found possible error is equivalent to a recommendation to revoke the underlying UPA. Baumgrass (2011) propose a process-centric methodology for refining existing RBAC states. They use event logs to derive RBAC artifacts (i.e. components of a role model like employee-role, role-role or role-permission assignments) to extend a role set. The authors stress that these artifacts are merely update candidates and must be integrated into the role model manually by a human decider, e.g. with the help of a role engineering tool. Hu et al. (2010a) present a tool which generates update steps that migrate an ACP set into a target state. Their tool can recommend different migration paths to a human, who has to decide which (if any) of them might be suitable. Rao et al. (2021), Han et al. (2012) and Chakraborty and Ray (2006) propose RBAC extensions which generate recommendations for updating user-role assignments. While Rao et al. (2021) and Han et al. (2012). propose rule-based approaches, Chakraborty and Ray (2006) propose to recommend user-role-assignments on the basis of users' trustworthiness.

It is noticeable that especially those publications that take a process perspective in optimizing ACPs require that optimizations are recommended to human deciders before they are implemented. Publications that propose algorithms for optimization often leave this requirement out. As a result, many optimization methods are not applicable for recommendation-based optimization. In order to be used for recommendation-based optimization, an optimization method must fulfill three requirements: (i) An optimization method needs to create individually decidable ACP update steps. This means that it must be possible to discard one step without having to discard the remaining ones. (ii) The decidable update steps must be small enough to be meaningfully decidable by a single decider. (iii) The decidable update steps must be human-understandable. The requirements (ii) and (iii) indicate that small update steps are preferable to large ones. Furthermore, it is helpful if the optimization method has a concept of the semantic structure of the ACP set. For example, ACPs that affect specific application systems can be treated and recommended in a bundle that is delegated to the respective system owner for decision. These requirements show that not every optimization algorithm can be reasonably adapted for recommendation-based optimization: For example, the output of a graph optimization algorithm that takes a role set as input and generates a single optimized role set cannot easily be converted into recommendations, since the decider would have to accept or

reject the entire optimized role set. Otherwise, if only one included update step were rejected, the remaining optimization would be incomplete and the resulting role set would likely be erroneous. In a real-world business environment with requirements for business continuity, change management processes and regulatory requirements, the ability to embed ACP optimization steps into the existing process landscape is a crucial factory for the practical applicability of an optimization method.

5.5. Limitations of this Work

The content of this study is limited by the selection of ACP quality and optimization criteria. The 16 quality criteria presented in section 3.2 provide a broad overview of the quality consideration of ACPs in the scientific IAM literature. Nevertheless, we do not claim completeness, as a representative summary would require a structured, reproducible literature survey. Moreover, the research scope of the literature survey is limited by the six selected optimization objectives defined in section 4.2.1. To ensure the relevance of the chosen criteria, we adopted the six optimization criteria developed by Beckerle and Martucci (2013) and confirmed them as central to ACPs' fitness for use in further literature research. Nevertheless, this represents a preselection that influences the literature unearthed in the survey and the findings based on it.

The literature research process of the survey also represents a possible limitation: We found that many publications use proprietary terminology, or aim at specific problems that are semantically equivalent to ACP maintenance or optimization, but not formulated as such (for example "ACP anomaly resolution"). The heterogeneity of the literature poses a challenge to structured literature research. Although we have adhered to the methodology presented in section 4.1, we cannot rule out with certainty that we have overlooked literature that fits the scope of the survey.

Due to its broad scope, this work cannot provide a detailed comparative evaluation of the various approaches that researchers proposed for optimization of ACPs. The analyzed publications differ in terms of their contributed research artifact, their grade of automation, and their optimization objective. It is difficult to compare a process model for ACP optimization with an algorithm for ACP optimization, or to compare an algorithm for reducing redundancies with an algorithm for correcting conflicts. We tried to address this challenge by formulating a generalized problem description, and by formulating analysis criteria are applicable among the heterogeneous literature base. Moreover, this work does not evaluate the analyzed approaches for their effectiveness or correctness, since a formal evaluation of the analyzed publications would have exceeded its scope.

Furthermore, the quality of ACPs is not well developed scientifically. While existing research proposes numerous proprietary definitions of *good* ACPs, there are few publications that address the topic holistically. The six optimization objectives on which the survey is based are well documented in the literature. Other quality-related ACP properties that are presented in section 3.2 are cited less frequently. Further groundwork on ACP quality is desirable.

6. Conclusion

This work studied the optimization of ACPs with the following contributions: (i) We show that the quality of ACPs constitutes an instance of the data quality concept as defined by Wand and Wang (1996) and provide a definition of ACP quality based on it. We give a broad overview of ACP quality as perceived in the IAM literature and provide a definition of ACP optimization. (ii) We present a structured literature survey that categorizes and analyzes existing methods for ACP optimization. We point out that the reduction of excessive and missing UPAs comes with the paramount challenge of *identifying* these inaccuracies. Once found, such inaccuracies can be corrected in a fully automated manner. Since identifying UPA inaccuracies relies heavily on context data, the availability of such data is a bottleneck. The reduction of complexity and redundancy in contrast are algorithmic problems which can be solved without any data other than the existing ACP set and its user-permission matrix. Approaches for conflict correction face the challenge of deciding whether to resolve a conflict by granting or permitting a particular UPA, and existing literature offers many strategies to address it. The issue of providing automation for role updating is thoroughly covered. Many approaches can be categorized into one of three classes: Rule-based automation, learning-based automation and trust-based automation. (iii) Building on the structured survey, we analyze important aspects of ACP optimization in more detail. In particular, we discuss three basic concepts for identifying UPA inaccuracies and their advantages and limitations. We analyze three prototypical types of data on which ACP optimization methods commonly rely and discuss their availability: While many ACP optimization methods require the existence of access logs, we point out that their availability may be limited in practice. Update logs, on the other hand, have not yet been precisely defined by researchers or incorporated into optimization methods. However, many authors emphasize their value as a possible source of information in optimizing ACPs. Furthermore, we analyze the concepts of minimal perturbation and recommendation-based optimization and argue their relevance for ACP optimization. Although both concepts are known to the research community, we point out that most optimization methods do not take them into account.

Future work has several starting points to contribute to the optimization of ACPs. (i) The quality of ACPs is often addressed in research, but it lacks theoretical foundations. There is no comprehensive literature survey that compiles existing ACP quality dimensions in a structured and reproducible process (with the exception of the quality criteria works named in section 3.2). Also, the correlations of quality dimensions, i.e. which dimension influences another positively, negatively or not at all, are not comprehensively studied. Furthermore, there is no model that analyzes which

ACP quality dimensions are subordinate or superordinate to others: For example, accuracy or maintainability seem to be aggregated dimensions that are based on different properties of ACPs. A structured overview, e.g. in the form of a topology, could provide clarity here and help standardize terminology in IAM research. (ii) Update logs are hardly developed as a data source. While especially the identification of UPA inaccuracies urgently needs other data sources than access logs, there is no definition of how update logs are structured or can be obtained in an IAM infrastructure. This groundwork could be used to develop methods for the (semi-)automated identification of UPA accuracies, which could be used to improve security and business continuity in organizations. (iii) While purely algorithmic or purely process-related optimization methods have been proposed by research, there is a lack of publications describing how algorithmic optimization can be embedded in existing process landscapes. Methodologies that implement optimization procedures, e.g. with tool support, could help to close this gap. Furthermore, practical reports or case studies would be a valuable aid for researchers trying to align theoretical ACP optimization methods with real-world needs. An analysis of the extent to which fully or semi-automated optimization is applicable in practical scenarios would also be helpful. (iv) Finally, there is very little research on access reviews: Although numerous organizations need to invest significant effort into the execution of access reviews on a regular basis, little research effort has been made to improve their limited effectiveness.

The optimization of ACPs is a relevant and ongoing research topic, and practitioners are left with major challenges that are yet to be solved. Existing research differs greatly in scope and terminology, which indicates that the research topic has yet to gain maturiy. We hope to contribute to its standardization with this work.

CRediT authorship contribution statement

Sascha Kern: Conceptualization, Methodology, Validation, Data Curation, Writing - Original Draft, Writing - Review & Editing, Project administration. **Thomas Baumer:** Data Curation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Sebastian Groll:** Conceptualization, Writing - Review & Editing. **Ludwig Fuchs:** Supervision. **Günther Pernul:** Writing - Review & Editing, Supervision, Funding acquisition.

Acknowledgement

The research leading to these results was supported by the German Federal Ministry of Education and Research as part of the DEVISE project (https://devise.ur.de/).

References

Aftab, M.U., Habib, M.A., Mehmood, N., Aslam, M., Irfan, M., 2015. Attributed role based access control model, in: 2015 Conference on Information Assurance and Cyber Security (CIACS), pp. 83–89. doi:10.1109/CIACS.2015.7395571.

Ahn, G.J., Sandhu, R., 2000. Role-based authorization constraints specification. ACM Trans. Inf. Syst. Secur. 3, 207–226. URL: https://doi.org/10.1145/382912.382913, doi:10.1145/382912.382913.

Al-Kahtani, M., Sandhu, R., 2002. A model for attribute-based user-role assignment, in: 18th Annual Computer Security Applications Conference, 2002. Proceedings., pp. 353–362. doi:10.1109/CSAC.2002.1176307.

Al-Kahtani, M., Sandhu, R., 2003. Induced role hierarchies with attribute-based rbac, in: Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 142–148. URL: https://doi.org/10.1145/775412.775430, doi:10.1145/775412.775430.

Al-Kahtani, M., Sandhu, R., 2004. Rule-based rbac with negative authorization, in: 20th Annual Computer Security Applications Conference, pp. 405–415. doi:10.1109/CSAC.2004.32.

Argento, L., Margheri, A., Paci, F., Sassone, V., Zannone, N., 2018. Towards adaptive access control, in: Data and Applications Security and Privacy XXXII. Springer International Publishing, pp. 99–109. URL: https://doi.org/10.1007/978-3-319-95729-6_7, doi:10.1007/978-3-319-95729-6_7.

Bauer, L., Cranor, L.F., Reeder, R.W., Reiter, M.K., Vaniea, K., 2009. Real life challenges in access-control management, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. p. 899–908. URL: https://doi.org/10.1145/1518701.1518838, doi:10.1145/1518701.1518838.

Bauer, L., Garriss, S., Reiter, M.K., 2011. Detecting and resolving policy misconfigurations in access-control systems. ACM Trans. Inf. Syst. Secur. 14. URL: https://doi.org/10.1145/1952982.1952984, doi:10.1145/1952982.1952984.

Baumgrass, A., 2011. Deriving current state rbac models from event logs, in: 2011 Sixth International Conference on Availability, Reliability and Security, pp. 667–672. doi:10.1109/ARES.2011.104.

Beckerle, M., Martucci, L.A., 2013. Formal definitions for usable access control rule sets from goals to metrics, in: Proceedings of the Ninth Symposium on Usable Privacy and Security, pp. 1–11.

Benedetti, M., Mori, M., 2018. Parametric rbac maintenance via max-sat, in: Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 15–25. URL: https://doi.org/10.1145/3205977.3205987, doi:10.1145/3205977.3205987.

Benedetti, M., Mori, M., 2019. On the use of max-SAT and PDDL in RBAC maintenance. Cybersecurity 2. URL: https://doi.org/10.1186/s42400-019-0036-9, doi:10.1186/s42400-019-0036-9.

Benkaouz, Y., Erradi, M., Freisleben, B., 2016. Work in progress: K-nearest neighbors techniques for abac policies clustering, in: Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control, Association for Computing Machinery, New York, NY, USA. p. 72–75. URL: https://doi.org/10.1145/2875491.2875497, doi:10.1145/2875491.2875497.

Bobba, R., Gavrila, S., Gligor, V., Khurana, H., Koleva, R., 2005. Administering access control in dynamic coalitions, in: Proceedings of the 19th Conference on Large Installation System Administration Conference - Volume 19, USENIX Association, USA. p. 23.

Calo, S., Manotas, I., de Mel, G., Cunnington, D., Law, M., Verma, D., Russo, A., Bertino, E., 2019. AGENP: An ASGrammar-based GENerative policy framework, in: Policy-Based Autonomic Data Governance. Springer International Publishing, pp. 3–20. URL: https://doi.org/10.1007/978-3-030-17277-0_1, doi:10.1007/978-3-030-17277-0_1.

Chakraborty, S., Ray, I., 2006. Trustbac: Integrating trust relationships into the rbac model for access control in open systems, in: Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 49–58. URL: https://doi.org/10.1145/1133058.1133067, doi:10.1145/1133058.1133067.

Chakraborty, S., Sandhu, R., 2021a. Formal analysis of rebac policy mining feasibility, in: Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy, Association for Computing Machinery, New York, NY, USA. p. 197–207. URL: https://doi.org/

10.1145/3422337.3447828, doi:10.1145/3422337.3447828.

Chakraborty, S., Sandhu, R., 2021b. On feasibility of attribute-aware relationship-based access control policy mining, in: Barker, K., Ghazinour, K. (Eds.), Data and Applications Security and Privacy XXXV, Springer International Publishing, Cham. pp. 393–405.

Cheminod, M., Durante, L., Seno, L., Valenza, F., Valenzano, A., 2017. Automated fixing of access policy implementation in industrial networked systems, in: 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS), pp. 1–9. doi:10.1109/WFCS.2017.7991947.

Cheminod, M., Durante, L., Seno, L., Valenza, F., Valenzano, A., 2019. A comprehensive approach to the automatic refinement and verification of access control policies. Computers & Security 80, 186–199. URL: https://www.sciencedirect.com/science/article/pii/S0167404818303870, doi:https://doi.org/10.1016/j.cose.2018.09.013.

Cheminod, M., Durante, L., Valenza, F., Valenzano, A., 2018. Toward attribute-based access control policy in industrial networked systems, in: 2018 14th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1–9. doi:10.1109/WFCS.2018.8402339.

Cheng, Y., Park, J., Sandhu, R., 2014. Attribute-aware relationship-based access control for online social networks, in: Atluri, V., Pernul, G. (Eds.), Data and Applications Security and Privacy XXVIII, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 292–306.

Cheng, Z., Royer, J.C., Tisi, M., 2019. Removing problems in rule-based policies, in: ICT Systems Security and Privacy Protection. Springer International Publishing, pp. 120–133. URL: https://doi.org/10.1007/978-3-030-22312-0_9, doi:10.1007/978-3-030-22312-0_9.

Colantonio, A., Di Pietro, R., Ocello, A., Verde, N.V., 2012. Visual role mining: A picture is worth a thousand roles. IEEE Transactions on Knowledge and Data Engineering 24, 1120–1133. doi:10.1109/TKDE.2011.37.

Cotrini, C., Weghorn, T., Basin, D., 2018. Mining abac rules from sparse logs, in: 2018 IEEE European Symposium on Security and Privacy (EuroS P), pp. 31–46. doi:10.1109/EuroSP.2018.00011.

Currey, J., McKinstry, R., Dadgar, A., Gritter, M., 2020. Informed privilege-complexity trade-offs in rbac configuration, in: Proceedings of the 25th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 119–130. URL: https://doi.org/10.1145/3381991.3395597, doi:10.1145/3381991.3395597.

Das, S., Sural, S., Vaidya, J., Atluri, V., 2018. HyPE: A hybrid approach toward policy engineering in attribute-based access control. IEEE Letters of the Computer Society 1, 25–29. URL: https://doi.org/10.1109/locs.2018.2889980, doi:10.1109/locs.2018.2889980.

Deng, F., Yu, Z., Liu, W., Luo, X., Fu, Y., Qiang, B., Xu, C., Li, Z., 2021. An efficient policy evaluation engine for xacml policy management. Information Sciences 547, 1105–1121. URL: https://www.sciencedirect.com/science/article/pii/S0020025520308148, doi:https://doi.org/10.1016/j.ins.2020.08.044.

Deng, F., Zhang, L.Y., 2017. Elimination of policy conflict to improve the PDP evaluation performance. Journal of Network and Computer Applications 80, 45–57. URL: https://doi.org/10.1016/j.jnca.2016.12.001, doi:10.1016/j.jnca.2016.12.001.

Dia, O.A., Farkas, C., 2012. A practical framework for policy composition and conflict resolution. International Journal of Secure Software Engineering 3, 1–26. URL: https://doi.org/10.4018/jsse.2012100101, doi:10.4018/jsse.2012100101.

Dos Santos, D.R., Westphall, C.M., Westphall, C.B., 2014. A dynamic risk-based access control architecture for cloud computing, in: 2014 IEEE Network Operations and Management Symposium (NOMS), IEEE. pp. 1–9.

Dunlop, N., Indulska, J., Raymond, K., 2003. Methods for conflict resolution in policy-based management systems, in: Seventh IEEE International Enterprise Distributed Object Computing Conference, 2003. Proceedings., IEEE. pp. 98–109.

Elliott, A., Knight, S., 2010. Role explosion: Acknowledging the problem., in: Arabnia, H.R., Reza, H., Deligiannidis, L., Cuadrado-Gallego, J.J., Schmidt, V., Solo, A.M.G. (Eds.), Software Engineering Research and Practice, CSREA Press. pp. 349–355. URL: http://dblp.uni-trier.de/db/conf/serp/serp2010.html#ElliottK10.

Fuchs, L., Broser, C., Pernul, G., 2009. Different approaches to in-house identity management - justification of an assumption, in: 2009 International Conference on Availability, Reliability and Security, pp. 122–129. doi:10.1109/ARES.2009.154.

Fuchs, L., Kunz, M., Pernul, G., 2014. Role model optimization for secure role-based identity management, in: European Conference on Information Systems (ECIS), pp. 1–15.

Fuchs, L., Pernul, G., 2007. Supporting compliant and secure user handling - a structured approach for in-house identity management, in: The Second International Conference on Availability, Reliability and Security (ARES'07), pp. 374–384. doi:10.1109/ARES.2007.145.

Fuchs, L., Pernul, G., 2008. HyDRo – hybrid development of roles, in: Information Systems Security. Springer Berlin Heidelberg, pp. 287–302. URL: https://doi.org/10.1007/978-3-540-89862-7_24, doi:10.1007/978-3-540-89862-7_24.

Fuchs, L., Pernul, G., 2010. Reducing the risk of insider misuse by revising identity management and useraccount data, in: 2nd Int. Workshop on Managing Insider Security Threats, Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA). Morioka, Iwate, Japan, pp. 14–28. URL: https://epub.uni-regensburg.de/15129/.

Fuchs, L., Pernul, G., Sandhu, R., 2011. Roles in information security – a survey and classification of the research area. Computers & Security 30, 748–769. URL: https://www.sciencedirect.com/science/article/pii/S016740481100099X, doi:https://doi.org/10.1016/j.cose.2011.08.002.

Gal-Oz, N., Gonen, Y., Yahalom, R., Gudes, E., Rozenberg, B., Shmueli, E., 2011. Mining roles from web application usage patterns, in: Trust, Privacy and Security in Digital Business. Springer Berlin Heidelberg, pp. 125–137. URL: https://doi.org/10.1007/978-3-642-22890-2_11, doi:10.1007/978-3-642-22890-2_11.

Gardiyawasam Pussewalage, H.S., Oleshchuk, V.A., 2017. Attribute based access control scheme with controlled access delegation for collaborative e-health environments. Journal of Information Security and Applications 37, 50–64. URL: https://www.sciencedirect.com/science/article/pii/S221421261730128X, doi:https://doi.org/10.1016/j.jisa.2017.10.004.

Gilbert, N., 2021. 31 crucial insider threat statistics: 2021 latest trends & challenges. URL: https://financesonline.com/insider-threat-statistics/.

Godik, S., Moses, T., 2003. URL: https://www.oasis-open.org/committees/xacml/repository/cs-xacml-specification-1.1.pdf.

Groll, S., Kern, S., Fuchs, L., Pernul, G., 2021. Monitoring access reviews by crowd labelling, in: Trust, Privacy and Security in Digital Business. Springer International Publishing, pp. 3–17. URL: https://doi.org/10.1007/978-3-030-86586-3_1, doi:10.1007/978-3-030-86586-3_1.

Guarnieri, M., Arrigoni Neri, M., Magri, E., Mutti, S., 2013. On the notion of redundancy in access control policies, in: Proceedings of the 18th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 161–172. URL: https://doi.org/10.1145/2462410.2462426, doi:10.1145/2462410.2462426.

Guo, Q., Vaidya, J., Atluri, V., 2008. The role hierarchy mining problem: Discovery of optimal role hierarchies, in: 2008 Annual Computer Security Applications Conference (ACSAC), pp. 237–246. doi:10.1109/ACSAC.2008.38.

Hadj, M.A.E., Benkaouz, Y., Freisleben, B., Erradi, M., 2017. ABAC rule reduction via similarity computation, in: Networked Systems. Springer International Publishing, pp. 86–100. URL: https://doi.org/10.1007/978-3-319-59647-1_7, doi:10.1007/978-3-319-59647-1_7.

Hadj, M.A.E., Erradi, M., Khoumsi, A., Benkaouz, Y., 2018a. Validation and correction of large security policies: A clustering and access log based approach, in: 2018 IEEE International Conference on Big Data (Big Data), pp. 5330–5332. doi:10.1109/BigData.2018.8622610.

Hadj, M.A.E., Khoumsi, A., Benkaouz, Y., Erradi, M., 2018b. Formal approach to detect and resolve anomalies while clustering ABAC policies. ICST Transactions on Security and Safety 5, 156003. URL: https://doi.

org/10.4108/eai.13-7-2018.156003, doi:10.4108/eai.13-7-2018.156003.

Hadj, M.A.E., Khoumsi, A., Benkaouz, Y., Erradi, M., 2019. Efficient security policy management using suspicious rules through access log analysis, in: Networked Systems. Springer International Publishing, pp. 250–266. URL: https://doi.org/10.1007/978-3-030-31277-0_16, doi:10.1007/978-3-030-31277-0_16.

Han, D.j., Zhuo, H.k., Xia, L.t., Li, L., 2012. Permission and role automatic assigning of user in role-based access control. Journal of Central South University 19, 1049–1056. URL: https://doi.org/10.1007/s11771-012-1108-0, doi:10.1007/s11771-012-1108-0.

Hein, P., Biswas, D., Martucci, L.A., Muhlhauser, M., 2011. Conflict detection and lifecycle management for access control in publish/subscribe systems, in: 2011 IEEE 13th International Symposium on High-Assurance Systems Engineering, pp. 104–111. doi:10.1109/HASE.2011.50.

Heinrich, B., Klier, M., 2009. A novel data quality metric for timeliness considering supplemental data, in: Newell, S., Whitley, E.A., Pouloudi, N., Wareham, J., Mathiassen, L. (Eds.), 17th European Conference on Information Systems, ECIS 2009, Verona, Italy, 2009, pp. 2651–2662. URL: http://aisel.aisnet.org/ecis2009/14.

Herzberg, A., Mass, Y., Mihaeli, J., Naor, D., Ravid, Y., 2000. Access control meets public key infrastructure, or: assigning roles to strangers, in: Proceeding 2000 IEEE Symposium on Security and Privacy. S P 2000, pp. 2–14. doi:10.1109/SECPRI.2000.848442.

Hill, L., 2006. How automated access verification can help organizations demonstrate HIPAA compliance: A case study. J Healthc Inf Manag 20, 116–122.

Horne, D., 2011. Permissions. Springer US, Boston, MA. chapter Permissions. pp. 924–927. URL: https://doi.org/10.1007/978-1-4419-5906-5_786, doi:10.1007/978-1-4419-5906-5_786.

Hornsteiner, M., Groll, S., Puchta, A., 2020. Towards a user-centric iam entitlement shop - learnings from the e-commerce, in: 13th International Conference on Security of Information and Networks, Association for Computing Machinery, New York, NY, USA. pp. 1–4. URL: https://doi.org/10.1145/3433174.3433585, doi:10.1145/3433174.3433585.

Hounder, F., 2010. Conflict Detection and Resolution of XACML Policies. Master's thesis. University of Applied Sciences Rapperswil.

Hu, H., Ahn, G.J., Kulkarni, K., 2011. Anomaly discovery and resolution in web access control policies, in: Proceedings of the 16th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 165–174. URL: https://doi.org/10.1145/1998441.1998472, doi:10.1145/1998441.1998472.

Hu, H., Ahn, G.J., Kulkarni, K., 2012. Detecting and resolving firewall policy anomalies. IEEE Transactions on Dependable and Secure Computing 9, 318–331. URL: https://doi.org/10.1109/tdsc.2012.20, doi:10.1109/tdsc.2012.20.

Hu, H., Ahn, G.J., Kulkarni, K., 2013. Discovery and resolution of anomalies in web access control policies. IEEE Transactions on Dependable and Secure Computing 10, 341–354. URL: https://doi.org/10.1109/tdsc.2013.18, doi:10.1109/tdsc.2013.18.

Hu, J., Khan, K.M., Zhang, Y., Bai, Y., Li, R., 2016. Role updating in information systems using model checking. Knowledge and Information Systems 51, 187–234. URL: https://doi.org/10.1007/s10115-016-0974-4, doi:10.1007/s10115-016-0974-4.

Hu, J., Zhang, Y., Li, R., 2010a. Towards automatic update of access control policy, in: Proceedings of the 24th International Conference on Large Installation System Administration, USENIX Association, USA. p. 1–7.

Hu, J., Zhang, Y., Li, R., Lu, Z., 2010b. Role updating for assignments, in: Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 89–98. URL: https://doi.org/10.1145/1809842.1809859, doi:10.1145/1809842.1809859.

Hu, V.C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K., 2014. Guide to Attribute Based Access Control (ABAC) Definition and Considerations. Technical Report. U.S. Department of Commerce. URL: https://doi.org/10.6028/nist.sp.800-162, doi:10.6028/nist.sp.800-162.

Huang, J., Nicol, D.M., Bobba, R., Huh, J.H., 2012. A framework integrating attribute-based policies into role-based access control, in: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 187–196. URL: https://doi.org/10.1145/2295136.2295170, doi:10.1145/2295136.2295170.

Hummer, M., Groll, S., Kunz, M., Fuchs, L., Pernul, G., 2018. Measuring identity and access management performance - an expert survey on possible performance indicators, in: Proceedings of the 4th International Conference on Information Systems Security and Privacy, SCITEPRESS - Science and Technology Publications. pp. 233–240. URL: https://doi.org/10.5220/0006557702330240, doi:10.5220/0006557702330240.

Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G., 2015. Advanced identity and access policy management using contextual data, in: 2015 10th International Conference on Availability, Reliability and Security, pp. 40–49. doi:10.1109/ARES.2015.40.

Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G., 2016. Adaptive identity and access management - contextual data based policies. EURASIP Journal on Information Security 2016. URL: https://doi.org/10.1186/s13635-016-0043-2, doi:10.1186/s13635-016-0043-2.

Jabal, A.A., Davari, M., Bertino, E., Makaya, C., Calo, S., Verma, D., Russo, A., Williams, C., 2019. Methods and tools for policy analysis. ACM Computing Surveys 51, 1–35. URL: https://doi.org/10.1145/3295749, doi:10.1145/3295749.

Jaferian, P., Rashtian, H., Beznosov, K., 2014. To authorize or not authorize: Helping users review access policies in organizations, in: Proceedings of the Tenth USENIX Conference on Usable Privacy and Security, USENIX Association, USA. p. 301–320.

Jin, C., Shen, A., Yu, W., 2016. The rbac system based on role risk and user trust. Int. J. Comput. Commun. Eng 5, 374–380.

Kern, A., Walhorn, C., 2005. Rule support for role-based access control, in: Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 130–138. URL: https://doi.org/10.1145/1063979.1064002, doi:10.1145/1063979.1064002.

Kunz, M., Fuchs, L., Hummer, M., Pernul, G., 2015a. Introducing dynamic identity and access management in organizations, in: Information Systems Security. Springer International Publishing, pp. 139–158. URL: https://doi.org/10.1007/978-3-319-26961-0_9, doi:10.1007/978-3-319-26961-0_9.

Kunz, M., Fuchs, L., Netter, M., Pernul, G., 2015b. How to discover high-quality roles? a survey and dependency analysis of quality criteria in role mining, in: Communications in Computer and Information Science. Springer International Publishing, pp. 49–67. URL: https://doi.org/10.1007/978-3-319-27668-7_4, doi:10.1007/978-3-319-27668-7_4.

Kunz, M., Puchta, A., Groll, S., Fuchs, L., Pernul, G., 2019. Attribute quality management for dynamic identity and access management. Journal of information security and applications 44, 64–79.

Levy, Y., Ellis, T.J., 2006. A systems approach to conduct an effective literature review in support of information systems research. Informing Sci. Int. J. an Emerg. Transdiscipl. 9, 181–212.

Lu, J., Xin, Y., Peng, H., Han, J., Lin, F., 2017. Supporting user authorization queries in RBAC systems by role-permission reassignment, in: Cyberspace Safety and Security. Springer International Publishing, pp. 468–476. URL: https://doi.org/10.1007/978-3-319-69471-9_35, doi:10.1007/978-3-319-69471-9_35.

Lu, J., Xu, D., Jin, L., Han, J., Peng, H., 2014. On the complexity of role updating feasibility problem in RBAC. Information Processing Letters 114, 597–602. URL: https://doi.org/10.1016/j.ipl.2014.06.003, doi:10.1016/j.ipl.2014.06.003.

Marouf, S., Shehab, M., Squicciarini, A., Sundareswaran, S., 2011. Adaptive reordering and clustering-based framework for efficient xacml policy evaluation. IEEE Transactions on Services Computing 4, 300–313. doi:10.1109/TSC.2010.28.

Meier, S., Fuchs, L., Pernul, G., 2013. Managing the access grid - a process view to minimize insider misuse risks, in: 11th International Conference on Wirtschaftsinformatik (WI2013), pp. 1051–1065. URL:

https://epub.uni-regensburg.de/27930/.

Menges, F., Latzo, T., Vielberth, M., Sobola, S., Pöhls, H.C., Taubmann, B., Köstler, J., Puchta, A., Freiling, F., Reiser, H.P., et al., 2021. Towards gdpr-compliant data processing in modern siem systems. Computers & Security 103, 102165.

Miseldine, P.L., 2008. Automated xacml policy reconfiguration for evaluation optimisation, in: Proceedings of the fourth international workshop on Software engineering for secure systems, pp. 1–8.

Mitra, B., Sural, S., Vaidya, J., Atluri, V., 2016. A survey of role mining. ACM Computing Surveys 48, 1–37. URL: https://doi.org/10.1145/2871148, doi:10.1145/2871148.

Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J., 2008. Mining roles with semantic meanings, in: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 21–30. URL: https://doi.org/10.1145/1377836.1377840, doi:10.1145/1377836.1377840.

Molloy, I., Chen, H., Li, T., Wang, Q., Li, N., Bertino, E., Calo, S., Lobo, J., 2010. Mining roles with multiple objectives. ACM Transactions on Information and System Security 13, 1–35. URL: https://doi.org/10.1145/1880022.1880030, doi:10.1145/1880022.1880030.

Moses, T., 2005. extensible access control markup language (xacml) version 2.0. URL: http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.

Narouei, M., Takabi, H., 2019. A nature-inspired framework for optimal mining of attribute-based access control policies, in: Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer International Publishing, pp. 489–506. URL: https://doi.org/10.1007/978-3-030-37231-6_29, doi:10.1007/978-3-030-37231-6_29.

Nazerian, F., Motameni, H., Nematzadeh, H., 2019. Emergency role-based access control (e-rbac) and analysis of model specifications with alloy. Journal of Information Security and Applications 45, 131–142. URL: https://www.sciencedirect.com/science/article/pii/S2214212618303843, doi:https://doi.org/10.1016/j.jisa.2019.01.008.

Ni, Q., Lobo, J., Calo, S., Rohatgi, P., Bertino, E., 2009. Automating role-based provisioning by learning from examples, in: Proceedings of the 14th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 75–84. URL: https://doi.org/10.1145/1542207.1542222, doi:10.1145/1542207.1542222.

Oberholzer, S., 2011. Optimizing xacml policies. University of Applied Sciences Rapperswil.

Pan, N., Sun, L., He, L.S., Zhu, Z.Q., 2018. An approach for hierarchical RBAC reconfiguration with minimal perturbation. IEEE Access 6, 40389–40399. URL: https://doi.org/10.1109/access.2017.2782838, doi:10.1109/access.2017.2782838.

Pang, C., Hansen, D., Maeder, A., 2007. Managing rbac states with transitive relations, in: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 139–148. URL: https://doi.org/10.1145/1229285.1229306, doi:10.1145/1229285.1229306.

Pang, C., Zhang, X., Zhang, Y., Ramamohanarao, K., 2008. The efficient maintenance of access roles with role hiding, in: Das, G., Sarda, N.L., Reddy, P.K. (Eds.), Proceedings of the 14th International Conference on Management of Data, December 17-19, 2008, IIT Bombay, Mumbai, India, Computer Society of India / Allied Publishers. pp. 139–149. URL: http://www.cse.iitb.ac.in/%7Ecomad/2008/PDFs/13.pdf.

Parkinson, S., Khan, S., Chrpa, L., 2020. Automated planning for administrating role-based access control. AAAI .

Puchta, A., Böhm, F., Pernul, G., 2019. Contributing to current challenges in identity and access management with visual analytics, in: Data and Applications Security and Privacy XXXIII. Springer International Publishing, pp. 221–239. URL: https://doi.org/10.1007/978-3-030-22479-0_12, doi:10.1007/978-3-030-22479-0_12.

Qi, H., Di, X., Li, J., 2018. Formal definition and analysis of access control model based on role and attribute. Journal of Information Security and Applications 43, 53–60. URL: https://www.sciencedirect.com/science/article/pii/S221421261730368X, doi:https://doi.org/10.1016/j.jisa.2018.09.001.

Rao, K.R., Nayak, A., Ray, I.G., Rahulamathavan, Y., Rajarajan, M., 2021. Role recommender-RBAC: Optimizing user-role assignments in RBAC. Computer Communications 166, 140–153. URL: https://doi.org/10.1016/j.comcom.2020.12.006, doi:10.1016/j.comcom.2020.12.006.

Royer, D., 2008. Enterprise identity management, in: Fischer-Hübner, S., Duquenoy, P., Zuccato, A., Martucci, L. (Eds.), The Future of Identity in the Information Society, Springer US, Boston, MA. pp. 433–446.

Saffarian, M., Tang, Q., Jonker, W., Hartel, P., 2009. Dynamic User Role Assignment in Remote Access Control. Number TR-CTIT-09-14 in CTIT Technical Report Series, Centre for Telematics and Information Technology (CTIT), Netherlands. Eemcs-eprint-15311.

Samarati, P., de Vimercati, S.C., 2001. Access control: Policies, models, and mechanisms, in: Focardi, R., Gorrieri, R. (Eds.), Foundations of Security Analysis and Design, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 137–196.

Sandhu, R.S., 1998. Role-based access control, in: Advances in Computers. Elsevier, pp. 237–286. URL: https://doi.org/10.1016/s0065-2458(08)60206-5, doi:10.1016/s0065-2458(08)60206-5.

Sandhu, R.S., Samarati, P., 1994. Access control: principle and practice. IEEE communications magazine 32, 40–48.

Schrimpf, A., Drechsler, A., Dagianis, K., 2021. Assessing identity and access management process maturity: First insights from the german financial sector. Information Systems Management 38, 94–115. URL: https://doi.org/10.1080/10580530.2020.1738601, doi:10.1080/10580530.2020.1738601, arXiv:https://doi.org/10.1080/10580530.2020.1738601.

Seifermann, S., Heinrich, R., Werle, D., Reussner, R., 2022. Detecting violations of access control and information flow policies in data flow diagrams. J. Syst. Softw. 184. URL: https://doi.org/10.1016/j.jss.2021.111138, doi:10.1016/j.jss.2021.111138.

Servos, D., Osborn, S.L., 2017. Current research and open problems in attribute-based access control. ACM Comput. Surv. 49. URL: https://doi.org/10.1145/3007204, doi:10.1145/3007204.

Shafiq, B., Vaidya, J.S., Ghafoor, A., Bertino, E., 2012. A framework for verification and optimal reconfiguration of event-driven role based access control policies, in: Proceedings of the 17th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 197–208. URL: https://doi.org/10.1145/2295136.2295172, doi:10.1145/2295136.2295172.

Shamoon, I., Rajpoot, Q., Shibli, A., 2012. Policy conflict management using xacml, in: 2012 8th International Conference on Computing and Networking Technology (INC, ICCIS and ICMIC), pp. 287–291.

Sheng, S., Osborn, S.L., 2004. A classifier-based approach to user-role assignment for web applications, in: Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 163–171. URL: https://doi.org/10.1007/978-3-540-30073-1_12, doi:10.1007/978-3-540-30073-1_12.

Stepien, B., Matwin, S., Felty, A., 2012. An algorithm for compression of xacml access control policy sets by recursive subsumption, in: 2012 Seventh International Conference on Availability, Reliability and Security, pp. 161–167. doi:10.1109/ARES.2012.38.

Strembeck, M., 2005. A role engineering tool for role-based access control, in: Proceedings of the Third Symposium on Requirements Engineering for Information Security SREIS, pp. 1–8.

Strembeck, M., 2010. Scenario-driven role engineering. IEEE Security & Privacy Magazine 8, 28–35. URL: https://doi.org/10.1109/msp.2010.46, doi:10.1109/msp.2010.46.

Takabi, H., Amini, M., Jalili, R., 2007. Trust-based user-role assignment in role-based access control, in: 2007 IEEE/ACS International Conference on Computer Systems and Applications, pp. 807–814. doi:10.1109/AICCSA.2007.370725.

Takabi, H., Joshi, J.B., 2010. Stateminer: An efficient similarity-based approach for optimal mining of role hierarchy, in: Proceedings of the 15th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 55–64. URL: https://doi.org/10.1145/1809842.1809853, doi:10.1145/1809842.1809853.

Tayi, G.K., Ballou, D.P., 1998. Examining data quality. Communications of the ACM 41, 54–57.

Tsiostas, D., Kittes, G., Chouliaras, N., Kantzavelou, I., Maglaras, L., Douligeris, C., Vlachos, V., 2020. The insider threat: Reasons, effects and mitigation techniques, in: 24th Pan-Hellenic Conference on Informatics, Association for Computing Machinery, New York, NY, USA. p. 340–345. URL: https://doi.org/10.1145/3437120.3437336, doi:10.1145/3437120.3437336.

Turkmen, F., Crispo, B., 2008. Performance evaluation of xacml pdp implementations, in: Proceedings of the 2008 ACM workshop on Secure web services, pp. 37–44.

Vaidya, J., Atluri, V., Guo, Q., Adam, N., 2008. Migrating to optimal rbac with minimal perturbation, in: Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 11–20. URL: https://doi.org/10.1145/1377836.1377839, doi:10.1145/1377836.1377839.

Verma, D., Calo, S., Witherspoon, S.A., Manotas, I., Bertino, E., Jabal, A.A., de Mel, G.R., Swami, A., Cirincione, G., Pearson, G., 2019. Managing training data from untrusted partners using self-generating policies, in: Pham, T. (Ed.), Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, SPIE. pp. 1–15. URL: https://doi.org/10.1117/12.2519682, doi:10.1117/12.2519682.

Wand, Y., Wang, R.Y., 1996. Anchoring data quality dimensions in ontological foundations. Communications of the ACM 39, 86–95.

Xia, H., Dawande, M., Mookerjee, V., 2014. Role refinement in access control: Model and analysis. INFORMS Journal on Computing 26, 866–884. URL: https://doi.org/10.1287/ijoc.2014.0603, doi:10.1287/ijoc.2014.0603.

Xiang, C., Wu, Y., Shen, B., Shen, M., Huang, H., Xu, T., Zhou, Y., Moore, C., Jin, X., Sheng, T., 2019. Towards continuous access control validation and forensics, in: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, Association for Computing Machinery, New York, NY, USA. p. 113–129. URL: https://doi.org/10.1145/3319535.3363191, doi:10.1145/3319535.3363191.

Xu, T., Naing, H.M., Lu, L., Zhou, Y., 2017. How do system administrators resolve access-denied issues in the real world?, in: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, Association for Computing Machinery, New York, NY, USA. p. 348–361. URL: https://doi.org/10.1145/3025453.3025999.

Xu, Z., 2014. Mining Meaningful Role-Based and Attribute-Based Access Control Policies. Ph.D. thesis. Stony Brook University.

Yi-qun, Z., Jian-hua, L., Quan-hai, Z., 2007. A general attribute based rbac model for web service, in: IEEE International Conference on Services Computing (SCC 2007), pp. 236–239. doi:10.1109/SCC.2007.8.

Zhang, D., Ramamohanarao, K., Ebringer, T., 2007. Role engineering using graph optimisation, in: Proceedings of the 12th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 139–144. URL: https://doi.org/10.1145/1266840.1266862, doi:10.1145/1266840.1266862.

Zhang, W., Chen, Y., Gunter, C., Liebovitz, D., Malin, B., 2013. Evolving role definitions through permission invocation patterns, in: Proceedings of the 18th ACM Symposium on Access Control Models and Technologies, Association for Computing Machinery, New York, NY, USA. p. 37–48. URL: https://doi.org/10.1145/2462410.2462422, doi:10.1145/2462410.2462422.

Zong, Y., Bhargava, B., Mahoui, M., Zhong, Y., 2011. Trustworthiness based authorization on www, in: IEEE Workshop on Security in Distributed Data Warehousing, pp. 1–6.

3 Monitoring Access Reviews by Crowd Labelling

Current status:	Published
Conference:	18th International Conference on Trust, Privacy and Security in Digital Business (TrustBus 2021), Virtual Event, September 27–30, 2021
Date of acceptance:	June 09, 2021
Full citation:	Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. Monitoring access reviews by crowd labelling. In Simone Fischer-Hübner, Costas Lambrinoudakis, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Trust, Privacy and Security in Digital Business (TrustBus)*, pages 3–17, Cham, 2021. Springer International Publishing.
Authors contributions:	Sebastian Groll 50% Sascha Kern 30% Ludwig Fuchs 10% Günther Pernul 10%

Conference Description: In the domain of digital businesses, concerns are raised regarding the lack of trust in electronic procedures and the extent to which information security and user privacy can be ensured. In answer to these concerns, the 17th International Conference on Trust, Privacy and Security in Digital Business (TrustBus'20) will provide an international forum for researchers and practitioners to exchange information regarding advancements in the state of the art and practice of trust and privacy in digital business.

Monitoring Access Reviews by Crowd Labelling

Sebastian Groll[1,2( )], Sascha Kern[1,2], Ludwig Fuchs[2], and Günther Pernul[1]

[1] University of Regensburg, Universitätsstraße 31, 93053 Regensburg, Germany
sebastian.groll@wiwi.uni-regensburg.de
[2] Nexis GmbH, Franz-Mayer-Straße 1, 93053 Regensburg, Germany

Abstract. Access reviews, i.e. the periodical security audit of access privileges, are a basic compliance and IT-security requirement for medium- and large-scale organizations. Assessing the quality of the reviewer's decisions ex-post can help to analyse the effectiveness of the measure and to identify structural or organizational shortcomings. Yet, current studies merely focus on improving the decision-making process itself. This paper develops a method for assessing the decision quality of access reviews by applying a solution from the crowd sourcing research realm. In order to achieve this, the problem of assessing decision quality of access reviews is generalized. It is shown that the abstract problem can be mapped to the problem of assessing the quality of crowd tagging decisions. Subsequently, an applicable solution of this research area is applied to access reviews. Furthermore, the selected approach is optimized to meet the specific challenges of access review data.

Keywords: Access reviews · Decision quality · Crowd sourcing · Identity and Access Management · Compliance

1 Introduction

Access reviews are a crucial task for Identity- and Access Management (IAM) and a basic compliance and IT security requirement for medium- and large-scale organizations [17,18]. While security concerns like insider threats represent an intrinsic motivation, external regulations and IT security standards[1] are an important driver for access reviews in practice. As access privileges are a subject to constant change, access reviews must be performed on a regular basis. They are typically executed by reviewers that hold responsibility for either an access privilege or an entity that it is assigned to. During an access review, a responsible human being manually inspects assignments of access privileges and decides if they are legitimate. If they are not, the privilege assignments are thereupon revoked. Note that access reviews are not used to evaluate the access

[1] Relevant examples are the Sarbanes-Oxley Act [2], Basel III [3], the European General Data Privacy Regulation [1], ISO 27002 [4] and the BSI Grundschutz [5].

S. Fischer-Hübner et al. (Eds.): TrustBus 2021, LNCS 12927, pp. 3–17, 2021.
https://doi.org/10.1007/978-3-030-86586-3_1

to resources in real-time but rather to inspect and certify an existing structure of access privilege assignments. The effective and efficient execution of access review campaigns remains a major challenge that is leveraged by technical and organizational complexity, time pressure and the sheer amount of review subjects [20]. In fact, large organizations may have numerous user accounts assigned to hundreds of thousands of access privileges spreading over dozens of application systems [13]. As a result, the manual inspection of all assignments, and the careful assessment of their legitimacy, are a time-consuming and error-prone procedure.

In the course of this work, we propose a generalized problem formulation for the quality definition of an access review decision. We then draw a link to the field of crowd sourcing and map the defined problem instance to the research area of crowd labelling. Subsequently, we adapt a well-known approach from existing literature for evaluating the quality of crowd labelling decisions and apply it to the evaluation of access review decisions. Consequently, we present a generalized data mining algorithm that can identify access review decisions with low quality.

In order to ensure both, relevance, and rigor, we follow the principles of the Action Design Research (ADR) methodology and continuously evaluate and improve our approach working together with practitioners on a real data set. The practitioners are responsible for managing the access review campaign rather performing the access reviews themselves. We cooperate with two companies, one IT security company with an expertise in IAM data analytics, and a large European company that performs company-wide access reviews on a regular basis. This work provides four major contributions: (1) We derive an abstract definition of an access review using a conceptual IAM model, (2) we show that research and methodologies of the crowd sourcing realm can successfully be applied to access review decision making and (3) adapt and improve a crowd labelling approach to access review quality assessment, thus contributing to both, the IT security and the crowd sourcing research realm. Finally, following the principles of ADR, (4) we evaluate and improve our approach by conducting a real-life project evaluation.

2 Theoretical Background

In the following we outline the most important definitions for our work in the IAM research area. Subsequently we give an overview of the relevant literature regarding IAM, access reviews and crowd labelling.

According to Pfitzman and Hansen [31] the term identity is a set or subset of attributes or characteristics of an entity that makes this entity uniquely identifiable among other entities. Relying on this definition, we define the term employee as referring to a uniquely identifiable real-world person working for a specific company. Following Pfitzman and Hansen [31], we use the definition of digital identity to derive the term account: An (user) account is unique in a specific application system and represents its employee in this context.

In order to manage the privilege assignments, a wide range of so-called access control models have been published. Sandhu et al. [33] propose the role-based

access control model (RBAC) inducing the role as intermediary between an employee and his privileges. Using roles can significantly reduce the number of assignments and therefore increase comprehensibility [14]. Fuchs and Preis [15] argue that there are multiple semantic types of roles like business roles, organizational roles or IT-roles[2]. Shortcomings of RBAC like the lack of fine-grained access control [6,36] or continuously increasing role sizes [26] and numbers [25] led to the development and improvement of attribute-based access control (ABAC) concepts [20,21].

The aim of access reviews is to inspect and certify the privilege assignments resulting from an implemented access control model. This might be done by reviewing certain permission assignments in an application or by reviewing conceptional assignments e.g. business roles which do not really exist on the application. Jaferian et al. [20] identify key challenges of access reviews and deal with the question of how to design an access review application in order to help the reviewers to make meaningful decisions. They design and implement a tool that is supposed to aid reviewers by enhancing the user interface. However, their focus lies on designing and evaluating the UI rather than evaluating the decision quality of the reviewers. Hill [17] presents a case study that showcases the introduction of software-supported access reviews and other access control management measures in a healthcare enterprise. Bobba et al. [7] discuss the design of tools that are supposed to aid domain administrators in the execution of access reviews. To the best of our knowledge, there are no scientific publications that explicitly cover the challenge of measuring access review decision quality in a structured manner.

Crowd sourcing is a collaborative and distributed problem-solving activity, where many workers join to solve various kinds of tasks. Examples for crowd sourcing projects are Galaxy Zoo[3], Amazon's Mechanical Turk[4], where users can publish tasks for crowdsourcing, or even collective projects like Wikipedia [24]. In Galaxy Zoo, for example, people work together in order to manually classify or tag different types of galaxies, like elliptical or spiral galaxies. Crowd sourcing comprises all sorts of tasks a community of crowd workers can perform.

In the following we focus on crowd labelling, which is a part of crowd sourcing. Crowd labelling is typically used to create labels for a large amount of data, e.g. to create input for supervised machine learning algorithms [16]. For example, above-mentioned Galaxy Zoo is a crowd labelling project. While crowd labelling has become a multi-million-dollar industry, the evaluation and improvement of crowd-sourced labels remains a crucial challenge (Lease 2011). The quality of labelling decisions is influenced by the labelers' expertise [27,30,38], their motivation [19,32], organizational circumstances and other factors [11,35]. As a result, crowd labelling may suffer from poor labelling quality or "spammers" [23] and thus numerous approaches to rate the quality of labels or crowd workers

[2] In the following we will refer to the term business role in order to emphasize the application- and technical- independent character of roles.
[3] https://www.zooniverse.org/projects/zookeeper/galaxy-zoo/.
[4] https://www.mturk.com/.

have been published [8,16,22,30]. Other approaches aim to improve the quality of labels in a generic way, e.g. by sorting out low quality labellers [10], or programmatically generating "gold labels" (i.e. correct labels that can be used to test the accuracy of labellers) [30].

2.1 Research Method

In order to design an ensemble artifact that addresses and satisfies organizational needs and ensures the rigor required for scientific contributions, we utilize the ADR method [34]. ADR relies on a close cooperation between researchers and practitioners and provides a framework for the dynamic interaction between the two parties. It defines four main stages containing principles to follow during their respective execution.

In the (1) Problem Formulation Phase the research opportunity is identified. The specific problem can be triggered by researchers, practitioners, or end-users. One crucial and challenging task is to define "the problem as an instance of a class of problems" [34]. This ensures that a solution for a more generalized problem is created and may extend the range of applicable theories. In this study the definition of a generalized problem enabled us to apply research approaches from the crowd sourcing realm on a problem of the IAM research area. Another principle of this phase is to design an artifact by utilizing theories. Finally, this phase also contains setting up an ADR team, which consists of practitioners and researchers, securing the long-term commitment of the practitioners as well as defining the roles and responsibilities of the team members. In the (2) Building, Intervention and Evaluation (BIE) phase the artifact is shaped in an iterative process. The practitioners and end-users implicitly take part in the design process by continuously evaluating the artifact. Moreover, the members of the ADR team benefit from their diverse expert knowledge. For example, our team consists of IAM researchers, IAM consultants and practitioners, respectively having different perspectives and insights on IAM and access reviews. While executing phase 1 and phase 2 the researcher simultaneously executes the (3) Reflection and Learning Phase. Thereby the researcher continuously transforms the gained knowledge to a broader class of problems. Therefore, our solution does not only improve the quality assessment of access reviews but is applicable for the broader field of crowd labelling as well. In the final phase (4) Formalization of Learning the researcher outlines the results achieved during the development of the artifact and its application to the organizational context. Due to the context-specific nature of ADR projects, a generalization of the concrete solution is necessary.

3 Problem Formulation

3.1 Practice-Inspired Research

For this study we worked together with two companies: The first one is a small IT security company, which offers tools and consulting services for mid-sized and

large organizations in the field of IAM. The provided services include the tool-based preparation and execution of access reviews. The second company is a large European organization with more than 15.000 employees. This company is forced to conduct company-wide access reviews on a regular basis due to existing compliance regulations. Our research was triggered by questions raised by customers of the partnering IAM security company like "How can I detect abnormalities in the access review process?", "How can I find out if the reviewers are doing their job carefully enough?" or "How can the correctness of decisions be ensured?". Thus, summarizing the requests and discussions we had with customers of the IT security company, we formulate the initial research question as follows: *"How can low-quality review decisions in an access review be identified ex-post?"*

Defining the roles of the practitioners [9], we decided that their responsibilities contain the provision of (anonymized) realworld IAM data (i.e. the access review data), the provision of contextual knowledge, as well as the regular evaluation of the developed artifact during the execution of the BIE cycle.

3.2 Defining Access Review Decision Quality as an Instance of a Class of Problems

According to Sein et al. [34], a "critical element of the problem formulation phase is defining the problem as an instance of a class of problems" [34]. This step is necessary to draw and contribute foundations and methodologies from and to the knowledge base. Thus, a funded and more general conceptual definition of the term "decision" in the context of access reviews. To achieve this, we adapt the existing conceptual IAM model proposed by Kunz et al. [26]. Their generic IAM model is a derivation of several relevant IAM standards (including RBAC and ABAC) and technologies containing and relating, amongst other things, employees, which are called digital identities in their work, accounts, permissions and business roles (Fig. 1)). Please note that Fig. 1 represents a simplified version of the original IAM model, specifically adapted to the requirements of representing access reviews and review decisions[5]. Employees refer to the real persons working for a company. They are represented by their accounts on the specific application systems of the organization. The accounts themselves are assigned to permissions representing fine-grained authorization objects to access services offered by these systems. Permissions may be structured hierarchically, i.e. owning a parent permission leads to the possession of additional child permissions. In contrast to permissions, business roles are less technical entities within IAM environments which are typically used to represent organizational or functional roles of employees like "Accountant" or "Sales Representative" [15]. They are used to bundle permissions from different application systems to grant employees access to all application systems needed for the daily work or for executing specific tasks or business processes. In contrast

[5] Therefore, we removed "Context" and "Policy" from the model because these entity types do not influence access reviews. We also renamed the term "(Digital) Identity" to "Employee" in order to fit our definitions and naming conventions, however, the meaning remains the same.

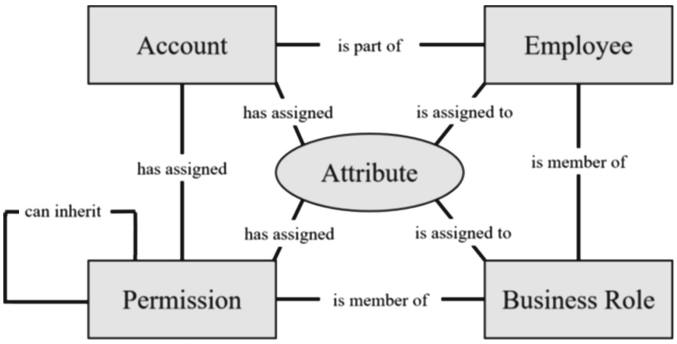


Fig. 1. Conceptual IAM model derived from Kunz et al. [26]

to accounts, business roles are directly assigned to employees, underlining their application-independent focus. Each of the entities described by Kunz et al. [26] may possess a set of attributes, which contains metadata of the respective entity. A typical attribute for an employee may be a job title or the location she is currently working at. Utilizing the conceptual IAM model of Kunz et al. [26] as well as our experiences from practical industry projects, we differentiate access reviews in two different main classes: (1) An entity itself, respectively its attributes, may be reviewed and (2) the relations between the entities, i.e. the edges in Fig. 1, may be reviewed. In the remainder of our practical analysis, we focus on the second class of access reviews (relationship reviews), however we argue that our research in general can be easily applied on the first class as well. Further generalizing, a decision d in an access review contains two entities (e.g. an account and a per-mission) and the decision whether these two entities should remain assigned to each other or not. Every entity e comprises a set of attributes x_e. Combined with the expert and contextual knowledge of the reviewer, these attributes form the foundation of the decision to be made. An access review decision by a reviewer r concerning the relation between two entities can further be expressed as $d_r(e1, e2) = \{x_{e1}, x_{e2}, y\}$ where y may be true or false, assuming the labels "keep relation" or "remove relation". Thus, an access review decision made by reviewer r comprises a set of features from the two entities and the information if these entities should remain linked or not. Finally, an access review with n decisions can be represented as a set $AR = \{d_i | i \in 1, ..., n\}$. Because we are especially interested in excessive assignments our research problem can be described as assessing the quality of every d_i with the label $y =$ "*keep relation*" to find suspicious and possibly wrong decisions.

4 Theory-Ingrained Artifact

According to the principles of ADR, we draw from the foundation of existing literature, to create the initial design of an artifact informed by theories. Therefore, we show that our generalized problem can be transferred to the research area of crowd sourcing and we apply selected approaches from this area. We also use well-established visualization techniques from IAM research to visualize our results.

The quality assessment problem of access reviews derived in the previous chapter can be mapped to the problem researchers are facing when assessing the

labelling quality of crowd workers. In an access review campaign, every decision $d(e1, e2)$ is a labelling task, with the labels "keep relation" and "remove relation". Like our previously defined vector of entity attributes x_e, the data which must be labelled in crowd labelling comprises several features and is typically expressed as a feature vector. And like one of the main problems of crowd labelling, our aim is to assess the quality of the decisions (i.e. labels).

Visualization is a key component of data analytics that can leverage the insights gained from otherwise less comprehensible data sets and make them accessible to non-technical experts. We suggest extending the access grid that was introduced by Meier, Fuchs and Pernul [28] for this sake as it is a well-proven interactive visualization technique that is tailored for displaying IAM data in an easily understandable manner. The reviewed entities are displayed as rows, and the referenced entities are displayed as columns. Each cell hence depicts the relationship between two entities: If an assignment exists between these two entities, the cell is filled; otherwise it is left blank. An assignment that exists, but was not subject of the review, is filled grey (i.e. assignments that deliberately were not part of the access review campaign). The color of a cell indicates a reviewer's decision: If an assignment was approved i.e. a positive decision was made, the cell is green; a negative decision is marked by a orange filling. If a positive decision is detected by our algorithm's outcome and therefore potentially has low quality, then a violet border is placed around the otherwise green cell (See Fig. 2).

4.1 Designing the Initial Artifact

To design the initial artifact, we utilize theories of the crowd sourcing domain and access grid visualizations to assess the quality of a set of given access review decisions. When mapping access reviews to crowd sourcing, there are several challenges, causing access reviews to be a more specific problem in the crowd sourcing domain: (1) Access reviews, as defined in the previous section, only consist of binary decisions, while crowdlabelling may have more than one possible label e.g. more types of galaxies to classify. (2) During access reviews, one decision is usually made by one reviewer (e.g. a responsible department owner of an employee or a responsible business role owner) whereas in typical crowd labelling tasks, several reviewers label the same data. Therefore, approaches like majority vote, which are often used in crowd labelling [8, 22, 37], cannot be applied easily. (3) Following the principle of least privilege [12, 33], the quality of positive decisions is more important in access reviews than the quality of negative decisions.

4.2 Applying Crowd Sourcing Theories to Access Reviews

Based on these assumptions, we selected the machine learning approach of Geva and Saar-Tsechansky [16] for our initial artifact design. It aims to create a ranking of decision makers based on their decision quality and is capable of handling binary decisions. One of its key assumptions is that every decision is created by exactly one reviewer. The underlying research question is: "how can we rank

workers by the relative quality of their decisions without resorting to the acquisition of additional and potentially costly, peer-review by other experts?" [16]. The key idea is to generate a so-called Pseudo Ground Truth (PGT) for every decision, i.e. an estimation of the correct y. This PGT can be compared with the decisions a decision maker has made to estimate her decision quality and to create a ranking of the decision makers. In order to create the PGT, a so-called base model is generated for every decision maker. The base model is trained with decision data, i.e. all decisions $d_r = \{x, y\}$ where r is the respective decision maker. These calculated base models can predict a decision for a given feature vector x, even if the decision maker, on whom the model is based, never made this decision. In other words, the base models are able to simulate reviewers' decisions. Following this concept, it is possible to create the PGT for a decision of a certain decision maker by conducting a vote with the base models of all the other decision makers. Note that the base models are necessary because of our assumption that every decision was only carried out by one decision maker and therefore it is not possible to use the decision data of the other decision makers directly. If a decision would have been carried out by multiple decision makers, a simulation using base models would not be necessary, because we could use the concrete decisions themselves. After being generated, the PGT is used to create a decision quality score for every decision maker by dividing the decisions, where the PGT and the decision maker agree, by the total amount of decisions that the decision maker has made. Geva and Saar-Tsechansky [16] claim that the precision of the decision quality score can further be improved by introducing the concept of confidence within the PGT and applying it as weight to every decision. To calculate the confidence value, the result of the vote from the different base models is used. Finally, the calculated decision quality scores of the decision makers can be used to create a ranking of decision makers regarding their decision quality. This general approach can be adopted to our problem: Throughout the process of an access review, the reviewer represents the decision maker according to the model of Geva and Saar-Tsechansky [16]. The feature vector x is represented by the attributes of the two entities that the reviewed assignment refers to, e.g. an account and a certain permission in an application system. The actual decision ("keep relation" or "remove relation") is the label y. Usually, several attributes for different entity types (e.g. employee, permission, etc.) are available in an identity management system and it is a critical and complex task to select a meaningful set of attributes. For example, the employee's gender usually has no influence on her permissions but the function she holds within the organization might heavily influence the assignment of certain permissions.

5 Building, Intervention and Evaluation

5.1 Mutually Influenced Roles

During the design and learning process, there was ongoing knowledge and feedback communication between the ADR team and the partnering IT security company. Firstly, the IT security company provided the data analysis platform

that enabled the ADR team to explore the customer data and to understand the review decisions in detail and as-a-whole. Secondly, the ADR team required contextual knowledge to understand individual review decisions in the real data set, and to determine ex-post whether a decision was erroneous or not. This context knowledge was acquired through workshops with responsible IT staff of both partnering companies. Information provided by the second industry partner regarding their organizational structures, attribute semantics, as well as responsibilities of reviewers proofed to be a valuable addition to the evaluation process.

5.2 Access Review Campaign Data

The industry partner executing the access review provided the ADR team with a real data set of a company-wide, quarterly access review campaign. The access review campaign was exclusively executed for account to permission assignments. The assignments were presented to the responsible reviewers in the form of employee to permission assignments to simplify their user interface and foster user acceptance. The delivered data set[6] comprised a total of 71.464 decisions regarding 10.891 employees and 1.623 permissions and was used both for the development of the artifact and for its evaluation. The provided decisions included 3.461 decisions that had no label (i.e. reviewer decision) asserted. This is in fact not unusual for an access review campaign of this scale due to organizational circumstances (e.g. illness of reviewers, technical challenges, etc.). Unlabelled decisions were omitted by our algorithm and did not affect the results. Table 1 shows the key figures of the data set. The ADR team used to work on various subsets of the data with a meaningful filter selection, i.e. decisions regarding employees of a single department and permissions of a single application system. This way, it was possible to use existing contextual knowledge, or to pose specific inquiries to representatives of the company to gather their expertise and to understand the origin of the observed decisions.

Table 1. Key data of the evaluation data sets.

Key data	Data set
Employees to review	10.891
Reviewers	855
Application systems	13
Permissions	1.623
Positive/Negative decisions	67.116 (93,9%)/887 (1,2%)
Not decided (reviewer did not make a decision)	3.461 (4,8%)
Total decisions to make	71.464

[6] The data set was gathered by using exports of the company's human resource system as well as exports of the different application systems.

5.3 Reciprocal Shaping

After implementing the initial artifact, the ADR team carried out the approach on the real data set in a first BIE cycle. However, the generated PGT comprised almost only positive decisions. Further analyzing the results highlighted three major reasons: Firstly, there was a very strong bias towards positive decisions in the data set with 93.9% of the provided decisions being positive, which was expected because most permission assignments are correct. The second discovered limitation is that a lot of the calculated base models had to make decisions that they were not qualified for. For example, a base model generated for a reviewer who had only made decisions regarding employees in the marketing department and permissions for application system "A" now was queried about a permission from application system "B" being assigned to an employee of the software development department. Such unqualified base models, which do not share any attributes with the currently processed decision, tend to guess a positive decision. The reason is, that most of the decisions used to generate the model are positive themselves. The third issue was noise that was generated by base models that were based on very few or even only one single decision. Their decision making was inflexible and used to overrule correct negative decisions just as unqualified base models did. As a result of these issues, the base models almost always voted for positive decisions, and a completely positive PGT was generated. To address these challenges, we extended the voting process of the base models in a way that the votes are no longer considered equally meaningful. The underlying assumption is that a base model is more successful in identifying errors correctly than another if it is based on more similar decisions.

We represent this qualification of a base model b(r), based on reviewer r, for a decision d by calculating an experience score $e_{d,b}$. Consequently, a decision made by this base model during the voting process is weighted with this experience. Let B be the set of all base models and B_d^+ be a subset of B with all base models that generate a positive label for decision d. Then the quality of a decision d^r made by reviewer r is the sum of the experience scores of B_d^+ , divided by the sum of the experience scores of B. The base model based on reviewer r itself is excluded from this score. The quality can then be expressed as:

$$q_{d,r} = \frac{\sum_{b^+ \in B_d^+ \setminus \{b(r)\}} (e_{d,b^+})}{\sum_{b \in B \setminus \{b(r)\}} (e_{d,b})}$$

We then define a quality threshold t that represents a minimum quality score which needs to be undercut to generate a finding for a suspected false decision. This way it is possible to adjust the output sensitivity of the algorithm. Formally, the decision d made by reviewer r is suspicious in terms of its quality, when $q_{(d,r)} < t, 0 \leq t \leq 1$ The quality threshold t can be adjusted to increase or decrease the amount of generated negative decisions and hence determines the detection sensitivity of the algorithm. While we argue that there is no determined optimum value as the amount of generated negative decisions depends on a data set's positive decision bias, we achieved good results with a threshold

of approximately 0.6[7]. As base model decisions are made based on the assigned entities' attributes x_{e1} and x_{e2}, we define that a base model has a higher experience than another if it has made more decisions on entities with equal attributes. We hence calculate a matching attribute score $attr_{d,b}$ for each base model decision by summing up all attributes of all decisions of the current base model, i.e. the decisions of the reviewer the base model is based on, that are equal to the current decision's attributes x_{e1}, x_{e2}. This matching attribute score is the basis of our introduced experience score. In order to normalize the sum of matching attributes, we used a sigmoid function, which is often used in artificial neural network research for similar normalization problems [29]. The sigmoid function ranks base models with few decisions very low, but still allows all base models with a certain "maturity" to be taken into consideration in the weighted vote. The experience score is hence defined as $e_{d,b} = \text{sigmoid}(attr_{d,b})$. With respect to the last ADR phase (Reflection and Learning) we emphasize that this improvement can be easily adopted to the generalized crowd labelling approach by simply enhancing the voting process of the base models with the method described in this chapter. Our approach can be helpful when dealing with data with an extreme bias. However, we argue that it might also be useful when dealing with non-biased data sets, as including the experience score decreases the influence of unqualified base models and increases the influence of qualified base models.

5.4 Evaluation

The final evaluation data set was selected by filtering the review campaign data to decisions regarding permissions of one large application system. It comprised 36.181 decisions with 34.744 positive, 383 negative and 1054 neutral labels. For the evaluation, three meaningful attributes were selected, and the quality threshold t was set to 0.6. The evaluation run yielded 33 results that indicated a false "keep relation" label. Manually investigating these findings together with our expert partners in the two companies, we were able to verify the correctness for 30 of them. This equates to an accordance of 90,9%. Please note that a higher quality threshold would have resulted in more findings and presumably also a lower accordance, while a lower threshold would have had the opposite effect. Using the selected strict parameterization and the low generated number of suspicious reviewer decisions allowed us to conduct focused review workshops with the industry partners without confronting them with a potentially large number of results (which in turn would have complicated the expert review process). Summing up, the chosen parameterization allowed us to yield a small number of likely correct results with an acceptably small error rate. As an example, Fig. 2 displays four findings discovered by the algorithm regarding two employees in an access grid. The employees are displayed in the rows of the visualization. Four positive decisions (see violet border) were made be the same reviewer, while

[7] Note that determining a suitable threshold must be done manually and the value highly depends on the imported data.

all other permissions of the two employees were marked for removal by another reviewer. Expert consultations revealed that the reviewer who made the negative decisions was the manager of these employees and knew that they had left the company during the past month. At the same time the positive decisions were made by a system administrator who was not informed about this fact. According to our industry partner's access review campaign experts, these findings are especially relevant, as they do not only expose possibly wrong decisions, but rather reveal structural and organizational problems concerning the quality of their overall IAM processes, e.g. lack in knowledge management or usability. Further expert evaluation revealed that reviewers had allegedly pressed a wrong button or decided too hastily.

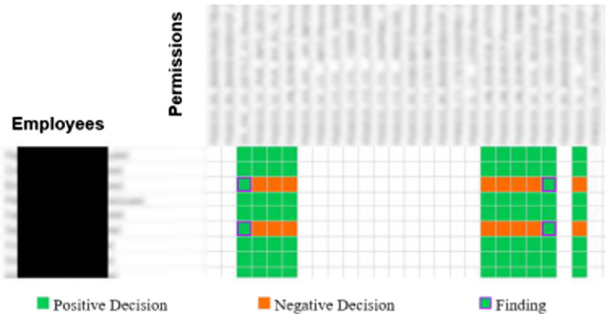


Fig. 2. Findings of the algorithm visualized with an access grid.

6 Conclusion

In the course of this work, we addressed the quality evaluation of access review decisions by answering the research question "How can low-quality review decisions in an access review be identified ex-post?". To achieve this, we firstly developed a generalized problem formulation. Subsequently, we mapped this problem and its characteristics to the research area of crowd sourcing. We facilitated an approach by Geva and Saar-Tsechansky [16] developed to assess the quality of crowd labelling decisions and consequently adapted it to the given problem formulation. Additionally, we developed a theory-ingrained artifact that allowed us to identify erroneous access review decisions automatically. During result evaluation we were able to prove the applicability of our approach for evaluating the quality of human access review decisions. We discussed interim results and considered the feedback provided by our industry partners in order to make sure that the research objectives are in line with actual practical needs. Following the ADR principle of generalized outcome, we provided both, a generalized problem, as well as a generalized solution formulation. The approach may be applied back to the original approach from the crowd labelling realm which, in turn, leads to the achievement of a general improvement and contribution to the field.

References

1. The European Parliament and the Council of the European Union: General Data Protection Regulation. https://eurlex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32016R0679. Accessed 22 Mar 2021
2. One Hundred Seventh Congress of the United States of America: "Sarabanes-Oxley-Act (SOX)". https://www.iso.org/standard/54533.html (2002). Accessed 22 Mar 2021
3. Basel committee on banking supervision: Basel III: a global regulatory framework for more resilient banks and banking systems. https://www.bis.org/publ/bcbs189.pdf (2010). Accessed 22 Mar 2021
4. International organization for standardization. "ISO/IEC 27002: Information technology - security techniques - code of practice for information security controls". https://www.iso.org/standard/54533.html (2013). Accessed 22 Mar 2021
5. Federal office for information security (BSI): "IT-Grundschutz". https://www.bsi.bund.de/EN/Topics/ITGrundschutz/itgrundschutz_node.html (2018). Accessed 22 Mar 2021
6. Azhar, A., Amin, M., Nauman, M., Shah, S.U.: Efficient selection of access control systems through multi criteria analytical hierarchy process. In: 2012 International Conference on Emerging Technologies, pp. 1–8 (2012). https://doi.org/10.1109/ICET.2012.6375419
7. Bobba, R., Gavrila, S., Gligor, V., Khurana, H., Koleva, R.: Administering access control in dynamic coalitions. In: Proceedings of the 19th Conference on Large Installation System Administration Conference, vol. 19, p. 23, LISA 2005. USENIX Association, USA (2005)
8. Brodley, C.E., Friedl, M.A.: Identifying mislabeled training data. J. Artif. Int. Res. **11**(1), 131–167 (1999)
9. Cole, R., Purao, S., Rossi, M., Sein, M.: Being proactive: where action research meets design research. In: ICIS 2005 Proceedings, p. 27 (2005)
10. Dekel, O., Shamir, O.: Vox populi: Collecting high-quality labels from a crowd. In: COLT (2009)
11. Erickson, L.B., Trauth, E.M., Petrick, I.: Getting inside your employees' heads: navigating barriers to internal-crowdsourcing for product and service innovation (2012)
12. Ferraiolo, D., Kuhn, D.R., Chandramouli, R.: Role-Based Access Control. Artech House, Boston (2003)
13. Fuchs, L., Pernul, G.: Supporting compliant and secure user handling-a structured approach for in-house identity management. In: The Second International Conference on Availability, Reliability and Security (ARES 2007), pp. 374–384. IEEE (2007)
14. Fuchs, L., Pernul, G.: HyDRo – hybrid development of roles. In: Sekar, R., Pujari, A.K. (eds.) ICISS 2008. LNCS, vol. 5352, pp. 287–302. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89862-7_24
15. Fuchs, L., Preis, A.: BusiROLE: a model for integrating business roles into identity management. In: Furnell, S., Katsikas, S.K., Lioy, A. (eds.) TrustBus 2008. LNCS, vol. 5185, pp. 128–138. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85735-8_13
16. Geva, T., Saar-Tsechansky, M.: Who's a good decision maker? data-driven expert worker ranking under unobservable quality (2016)

17. Hill, L.: How automated access verification can help organizations demonstrate HIPAA compliance: a case study. J. Healthcare Inf. Manag. **20**(2), 116 (2006)
18. Hummer, M., Kunz, M., Netter, M., Fuchs, L., Pernul, G.: Adaptive identity and access management-contextual data based policies. EURASIP J. Inf. Secur. **2016**(1), 1–16 (2016)
19. Ihl, A., Strunk, K.S., Fiedler, M.: The influence of utilitarian and hedonic motivation on success in crowd work (2018)
20. Jaferian, P., Rashtian, H., Beznosov, K.: To authorize or not authorize: helping users review access policies in organizations. In: 10th Symposium on Usable Privacy and Security ({SOUPS} 2014), pp. 301–320 (2014)
21. Jin, X., Krishnan, R., Sandhu, R.: A unified attribute-based access control model covering DAC, MAC and RBAC. In: Cuppens-Boulahia, N., Cuppens, F., Garcia-Alfaro, J. (eds.) DBSec 2012. LNCS, vol. 7371, pp. 41–55. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31540-4_4
22. Khattak, F.K., Salleb-Aouissi, A.: Robust crowd labeling using little expertise. In: Fürnkranz, J., Hüllermeier, E., Higuchi, T. (eds.) DS 2013. LNCS (LNAI), vol. 8140, pp. 94–109. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40897-7_7
23. Kittur, A., Chi, E.H., Suh, B.: Crowdsourcing user studies with mechanical Turk. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 453–456 (2008)
24. Kittur, A., et al.: The future of crowd work. In: Proceedings of the 2013 Conference on Computer Supported Cooperative Work, pp. 1301–1318 (2013)
25. Kuhn, D.R., Coyne, E.J., Weil, T.R.: Adding attributes to role-based access control. Computer **43**(6), 79–81 (2010)
26. Kunz, M., Puchta, A., Groll, S., Fuchs, L., Pernul, G.: Attribute quality management for dynamic identity and access management. J. Inf. Secur. Appl. **44**, 64–79 (2019)
27. Leicht, N., Rhyn, M., Hansbauer, G.: Can Laymen outperform experts? The effects of user expertise and task design in crowdsourced software testing (2016)
28. Meier, S., Fuchs, L., Pernul, G.: Managing the access grid-a process view to minimize insider misuse risks (2013)
29. Menon, A., Mehrotra, K., Mohan, C.K., Ranka, S.: Characterization of a class of sigmoid functions with applications to neural networks. Neural Netw. **9**(5), 819–835 (1996)
30. Oleson, D., Sorokin, A., Laughlin, G., Hester, V., Le, J., Biewald, L.: Programmatic gold: Targeted and scalable quality assurance in crowdsourcing. In: Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence. Citeseer (2011)
31. Pfitzmann, A., Hansen, M.: A terminology for talking about privacy by data minimization: anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management (2010)
32. Rouse, A.C.: A preliminary taxonomy of crowdsourcing (2010)
33. Sandhu, R.S., Coynek, E.J., Feinsteink, H.L., Youman, C.E.: Role-based access control models. Computer **29**(2), 38–47 (1996)
34. Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R.: Action design research. MIS Q. **35**(1), 37–56 (2011)
35. Tavanapour, N., Bittner, E.A.: The collaboration of crowd workers (2018)
36. Valecha, R., Kashyap, M., Rajeev, S., Rao, R., Upadhyaya, S.: An activity theory approach to specification of access control policies in transitive health workflows (2014)

37. Whitehill, J., Wu, T.F., Bergsma, J., Movellan, J., Ruvolo, P.: Whose vote should count more: optimal integration of labels from labelers of unknown expertise. Adv. Neural Inf. Process. Syst. **22**, 2035–2043 (2009)
38. Wöhner, T., Köhler, S., Peters, R.: Good authors = good articles?-how Wikis work. In: Wirtschaftsinformatik, pp. 872–886 (2015)

4 Separation of Duty in Information Security

Current status:	Revision
Journal:	*Submitted to*: ACM Computing Surveys
Date of acceptance:	-
Full citation:	Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Separation of duty in information security. *Submitted to: ACM Computing Surveys*.
Authors contributions:	Sebastian Groll 80% Ludwig Fuchs 10% Günther Pernul 10%

Conference Description: The primary purpose of the ACM Computing Surveys is to present new specialties and help practitioners and researchers stay abreast of all areas in the rapidly evolving field of computing. Computing Surveys focuses on integrating and adding understanding to the existing literature by publishing surveys and tutorials. The journal is published 9 times per year.

Separation of Duty in Information Security

SEBASTIAN GROLL, Universität Regensburg, Germany
LUDWIG FUCHS, Nexis GmbH, Germany
GÜNTHER PERNUL, Universität Regensburg, Germany

CCS Concepts: • **General and reference** → *Surveys and overviews*; • **Security and privacy**; • **Applied computing** → *Business process management*; *Enterprise computing*;

Additional Key Words and Phrases: Separation of Duty, Identity and Access Management, Workflows

1 INTRODUCTION

Separation of Duty (SoD) is a fundamental principle in information security. In simple words, SoD ensures that certain crucial steps of a process are carried out by different users. There are three important drivers for SoD: The first one is security, because SoD is an efficient measure to prevent fraud [19]. Consider a product ordering process in a company: first a certain product is ordered, later it is received and inspected and at the end the product is paid for. Every step is documented, but fraud is possible when a user provides false information. The impostor could fake to order goods, document the receipt and pay for them, even if the goods were never received. Several similar cases exist in the real world, like fraud in the finance and banking sector [25] or the the Wirecard case[1]. When SoD is successfully applied, it ensures that process steps are carried out by different persons, making this kind of fraud more difficult, unless these persons conspire [25].

Another driver for SoD is to prevent users from making process errors, violating policies or even inadvertently committing crimes. For example, in a financial institute, SoD can ensure that a certain loan contract is signed by two or more people. This may be required for the contract to be legally binding. A third driver for SoD, especially for large or highly regulated organizations, is to comply with legal requirements like the Sarbanes-Oxley Act [65], the German BSI Grundschutz [66] or the American Health Insurance Portability and Accountability Act [101].

The topic of SoD is a multifaceted subject that includes technical and organizational challenges on various levels. For instance, on a technical level, factors such as the access to data in databases, user permissions, and the applied access model must be considered. On an organizational level, workflows, tasks, and overall policy management are important aspects (among others). Therefore, over the last decades, a large stream of SoD literature, spanning over different research realms, has emerged. To the best of our knowledge and despite the practical relevance and the extensive research in the area, this stream of literature has never been structured nor comprehensively summarized. In order to close this research gap we conduct a systematic literature review[2], develop a classification scheme and analyse the

[1] https://www.reuters.com/business/finance/former-wirecard-boss-goes-trial-fraud-scandal-that-rocked-germany-2022-12-08/ accessed on 06-10-2023
[2] In the sense of "Stand-alone literature review" described in [67].

Manuscript submitted to ACM

state of the art as well as current trends in the overall SoD literature. Because this has never been done before, we are considering the scientific works from 1990 until now (2023).

2 FUNDAMENTALS

Separation of Duty, sometimes also referred as "Segregation of Duty", can be defined as an information security principle with the goal to create an internal control system that reduces the threat of insider fraud by dividing critical tasks, functions, and (access) privileges upon multiple users. Depending on the focus of the respective work, research looks at the topic of SoD from different perspectives. For instance, when focusing on access control, SoD is usually seen as a set of rules that restrict somehow the assignment of (technical) roles or permissions. On the other hand, looking on workflows, SoD defines tasks that must not be executed by the same person.

2.1 Early Research and Subsequent Progress

In research, the Clark–Wilson model for data integrity [19] may be considered as first strong impetus towards the area of SoD. In the formal model the list of transactions each user can perform on data objects, has to be manually certified to meet the SoD principle. By doing so, they implicitly propose access control and a set of restricted combinations of access permissions as means to enforce SoD in information systems.

The original Clark–Wilson model was considered as too laborious due to the manual effort when enforcing it [7, 89]. Also, the restriction that a user is not allowed to have conflicting permissions at all, which is referred to Static SoD, was considered a shortcoming regarding real world requirements. For example, when applying Static SoD to the product ordering process mentioned above, one user would always be entitled for ordering, a second one for inspecting the product and a third one for paying for it. It would never be possible, that the three persons change their roles when ordering another product. In their work on transaction control expressions and SoD, Sandhu et al. [7, 79, 81] suggest a more flexible SoD property called Dynamic SoD, which is enforced at runtime.

The necessity for Dynamic SoD is also emphasized by Nash and Poland [62] claiming that Static SoD lacks the possibility to exclude specific combinations of users, transactions, and data objects. Consequently, more SoD research was carried out in the access control research area. Particularly, the introduction of the role-based access control model (RBAC) [80], in which roles are set as an intermediary between the users and the permissions, extended the point of view from conflicting permission assignments to mutual exclusive roles. Further research exceeded this area, spanning over various topics and models like RBAC, the attribute-based access control model (ABAC), BPMN, UML, workflows, Petri nets, etc. as well as different algorithms or management approaches. Therefore, the need to structure these approaches becomes apparent.

2.2 Introducing SoD Properties

A first way to structure SoD research is provided by the definition of SoD properties [30, 89]. In the following, we informally summarize the commonly stated SoD properties[3]. On a high level, the properties are divided into Static SoD and Dynamic SoD containing Simple Dynamic SoD, Object Based SoD, Operational Based SoD and History Based SoD. Below, we summarize the SoD properties provided and defined by Simon and Zurko [89]:

[3]There are also less frequently mentioned SoD properties (e.g. Fuzzy SoD [95]).

- **Static SoD**: In a Static SoD rule, a user is not allowed to possess conflicting assignments[4] at all. This is the most restrictive form of SoD and therefore often denoted as impractical. The Dynamic SoD properties are less restrictive and additionally take the current context into account when being enforced.
- **Simple Dynamic SoD**: The Simple Dynamic SoD property allows a user to have conflicting assignments, but the user is not allowed to activate conflicting assignments at the same time.
- **Object Based SoD**: The Object Based SoD property allows the user to have conflicting assignments and also allows the user to activate those assignments at the same time. However, the user is not allowed to perform operations with conflicting assignments on the same object multiple times. E.g., a user might submit and approve loan applications, but she is not allowed to submit and approve the same loan application.
- **Operational Based SoD**: In Operational Based SoD a user can have conflicting assignments, but those assignments are not allowed to cover a complete business task. Therefore, there is always more than one user required to complete a certain business task.
- **History Based SoD**: Both, Object Based and Operational Based SoD, have some shortcomings. A user can either not perform two possibly harmless actions on an object twice or she is not able to perform all the actions of one business task on different objects. The History Based SoD property, as a combination of both, overcomes these shortcomings. A user is allowed to have restricted assignments, and the assignments may also cover a complete business process, however, the user is not allowed to perform all actions of a business process for one object with these assignments.

A more formalized and extended summary of SoD properties is provided by Virgil and Gligor [30]. They further extend the set of SoD properties in different ways, for example by adding static and dynamic properties to the Object Based SoD and the Operational Based SoD property. In total they define 11 different properties. Some of them remain theoretical in nature because they are even stricter than the Static SoD property.

However, using SoD properties as a classification scheme for existing literature and approaches has several drawbacks: Some literature treats SoD in a more coarse-grained way so that a matching SoD property cannot be determined. Other works treat several properties at once, and even at a different level of detail. Additionally, there is literature that cannot be assigned to any SoD property at all, because it exceeds the scope of classification. This refers to work regarding high-level SoD or management, for example. Finally, SoD properties were originally never intended to structure literature or approaches, but to describe how a specific SoD approach works. Therefore, the development of a more suitable classification scheme becomes necessary.

2.3 Key Terminology

Various publications tend to use different terminologies for the same concepts. In order to create a consistent understanding of the subject matter throughout this work, we are going to use the standardized terminology below for the most relevant entities:

- **User**: refers to a human person interacting with a system. In this study the user is typically the subject to which SoD policies are applied and member of an organization. In contrast to the human person, accounts are the digital representation of users in a specific application. Terms used in other work: *employee*, *worker*, *actor*, *insider*.

[4]Note that the work of [30, 89] is situated in the area of RBAC. We consciously use the term 'assignment' instead of 'role' to demonstrate broader applicability. However, work in other areas (e.g., workflows, BPMN) often builds upon the usage of the RBAC model to define authorizations.

- **Organization**: An association of people who, in our case, have to comply with SoD rules. Terms used in other work: *company, enterprise, corporation.*
- **Permission**: refers to a single, assignable permission in an application. Terms used in other work: *privilege, entitlement, access right.*
- **Role, technical role**: The role can be seen as a function, a user has in an organization. If subject to SoD a user might not be allowed to take two conflicting roles in an organization. On a more technical level, a role can be seen as a container of permissions a user gets when assigned to a role. The concept of applying a role model based on three layers (roles, technical roles, permissions) is widely adopted in real-life scenarios.

3 METHODOLOGY

In order to ensure the rigor of our research we utilized several methodical guidelines [49, 72, 99, 106] and adopted the Eight-Step Guide of Okoli and Schabram [67] as an overall process model to conduct a systematic stand-alone literature review. The execution of the steps and phases is briefly described below, but for a comprehensive insight into the procedure we recommend reading the guide. The first phase is concerned with planning the review. This contains defining the purpose of the study and developing appropriate research objectives as well as creating a research protocol and train the reviewers. We define the purpose of our study as follows:

To analyse the current stream of literature in order to synthesise the challenges, solutions, and state of the art for specifying, implementing or managing SoD.

Consequently, the following research objectives (RO1-RO3) were derived:

- **RO1**: Identify the relevant literature in the field of SoD.
- **RO2**: Structure the body of relevant literature in the field of SoD.
- **RO3**: Elaborate on the state of the art and identify the current challenges in the field of SoD.

After defining the research objectives, the next step in the research protocol is to define the process of discovering relevant literature. Therefore, we started our research with an extensive database search. The selected search terms are "Separation of Duty", "Separation of Duties", "Segregation of Duty" and "Segregation of Duties". These more general search terms are selected in order to cover all relevant literature. Note that they, however, yield in significantly more results than specific ones like "Separation of Duty Implementation". Depending on the respective database, the syntax of the search query was adjusted accordingly. The used databases are summed up with their results in Table 1. The meta data as well as the full text of the paper were included in the search. After the query execution, appropriate research publications were filtered using the following three step filter model: The first filter step is called Practical Screen and contains hard criteria so that the results are reproducible:

- Publication is a scientific contribution and written in English language
- Publication was published between 1990 and 2023
- Publication is a full paper in a journal or conference

In the next step (Quality Appraisal), the content of the publication needs to be screened in order to ensure that the publication focuses on SoD or at least contributes to the research area. For example, publications proposing a new access control model and show in a subsection how it can implement SoD are not included in the study, because there is no direct contribution to the research area. Note that one major challenge is to decide upon the inclusion of borderline

Table 1. Amount of publications found in the databases

Database	After query execution	After filtering	Website
ACM	747	17	https://dl.acm.org
AISeL	152	1	https://aisel.aisnet.org
IEE	182	32	https://ieeexplore.ieee.org
Science Direct	696	10	https://www.sciencedirect.com
Springer	1.394	23	https://link.springer.com
Cross search	-	5	
Total	3.171	88	

publications in this work. In case the initial screen of the paper did not lead to a clear decision, the following additional filter-criteria were used:

- The publication uses a key term[5] in the meta-data
- A key term occurs very frequently in the content
- Relevant SoD literature is cited
- SoD is introduced or treated exclusively in an own section or subsection

If at least three out of these four criteria are fulfilled, then a borderline paper is added to the result set.

Executing the search terms in the selected databases yielded in a total result set of 3.171 publications, while the manual filtering process led to a relevant result set of 83 publications. A consecutive forward and backward search was executed, yielding in 5 additional publications that were included in the result set. This low number further underlines the comprehensiveness of the database search.

During the filtering process and throughout multiple development cycles, a multi-level classification scheme for the relevant literature was developed based on the content of the analysed publications. The classification scheme, its limitations, and exclusions are presented in the next section. For transparency reasons, the results of the query execution as well as the final results and categorization are available on github[6].

4 CATEGORIZING THE CURRENT STATE OF THE ART OF SEPARATION OF DUTY LITERATURE

The first level of our categorization for SoD literature contains the four primary categories "SoD Specification", "SoD Enforcement", "SoD Analysis", and "High-level SoD and Management". The overall classification scheme can be viewed in Figure 1. Note that the scheme is based upon the thorough analysis of all included research publications. Depending on the desired level of diversification of categories, a more fine-grained or more course-grained classification would have been possible. Based on our analyses, the proposed scheme appears to be a very effective way to structure SoD research. The categories and sub-categories are further described in the following.

4.1 Introducing a Classification Scheme for SoD Literature

The primary category **SoD Specification** deals with the challenge of adapting and extending existing approaches in research in order to support the specification of SoD. It also addresses the issue of how SoD policies are structured. The addressed approaches that are extended with SoD features, can roughly be divided into models and modelling

[5] As key term several words and their abbreviations were defined like: "Separation of Duty" ("SoD"), "Segregation of Duty", "mutual exclusive roles" ("MER") etc.
[6] https://github.com/sgroll/sod_survey/

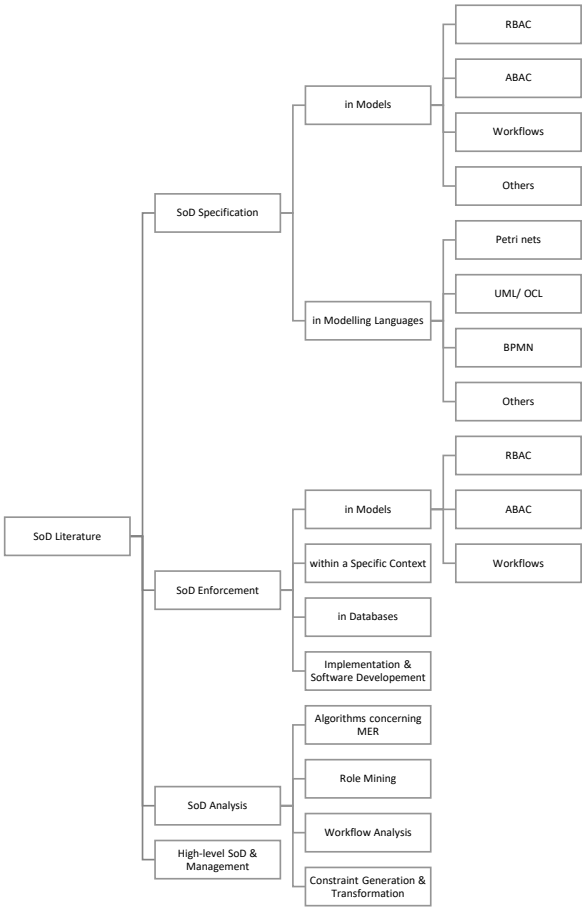


Fig. 1. A classification scheme for SoD literature

languages. Regarding models, e.g., the two prominent access control models RBAC and ABAC are covered. Additionally, work focusing on workflows and petri nets (PN) are part of this research category. The modelling languages, in turn, can be divided into UML/OCL and BPMN. In order to categorize approaches that are not directly assignable to a more generic term (e.g. the specification of SoD rules in information flows [28] or an algebra for expressing SoD rules [51]) the subcategory "others" is defined respectively.

The primary category **SoD Enforcement** deals with the challenge of implementing and enforcing SoD policies and rules. SoD Enforcement refers to ensuring that no SoD rules are violated. Naturally, papers that deal with SoD specification or SoD analysis may also include some discussion of how the specified SoD rules can be enforced. However, the SoD Enforcement category includes papers that especially focus on enforcement, e.g. when it is examined under complex circumstances. One subcategory handles SoD Enforcement in Models, and the identified literature can be

further divided into the subcategories RBAC, ABAC, and Workflows. For instance, literature that is assigned to the category enforcement in the RBAC model deals with the challenge of avoiding or identifying SoD violations in an RBAC state. The next subcategory, SoD Enforcement within a Specific Context, contains literature that focuses on enforcing SoD under various circumstances and conditions. This includes decentralized environments or SoD rules with spatial or time-based conditions. The last two subcategories for SoD Enforcement are SoD Enforcement in Databases, and Implementation and Software Development. The latter contains literature proposing prototypical implementations of e.g. SoD Enforcement engines as well as approaches of integrating SoD in the software development process.

The primary category **SoD Analysis** contains literature proposing algorithmic approaches to analyse, evaluate, simulate or transform SoD rules. One subcategory handles algorithms concerning MER policies. In contrast to the enforcement of SoD policies in the RBAC model, these algorithms aim at analyzing or optimizing role-assignments while staying compliant with MER constraints. The next category covers role mining in the context of SoD policies. Role mining refers to the automatic generation of roles usually by analyzing and grouping the current user-permission assignments. While a large number of role mining approaches exist in literature, only a very few address the topic of mining SoD compliant roles (see Table 2). The subcategory Workflow Analysis contains literature with algorithms that handles SoD in context with workflows. For example, an algorithm can generate a plan specifying which workflow steps have to be done by which user in order to stay compliant. Note that we explicitly exclude research that focuses on the workflow satisfiability problem (see section 4.2 for detailed explanation). The subcategory Constraint Generation and Transformation deals with algorithms that transfer a set of constraints to another set of constraints on e.g. a different granularity level. For example, when constraints are specified on the basis of permissions, the enforcement might be computational complex and a transformation to the role level might be necessary.

The primary category **High-level SoD and Management** contains literature that covers SoD on an organizational or management level. Literature in this category addresses challenges like how to introduce SoD in an organization in general, how to derive meaningful SoD rules, or how to assess the SoD maturity level of an organization.

4.2 Limitation and Exclusions

Due to the extensive size of initially gathered results and to keep the main focus of this work, some boundaries were defined: Vertical fragmentation, i.e. the separation of data over different cloud databases, is sometimes referred to as "Separation of Duties" [13]. The key idea is, that some data might only be sensitive when observed as a whole, and therefore, can be stored in a secure manner, when split between different databases. However, this concept does not match the semantic meaning of SoD treated in our work. We focus on the separation of responsibilities rather then a way to split up data in a secure manner.

A topic, which is closely related to SoD but explicitly excluded in this study is the workflow satisfiability problem [31] (WSP). The WSP deals with scenarios when a workflow, which is subject to restrictions, can not be successfully executed due to those restrictions. The cause for non-satisfiability could be any restriction, but usually SoD is used as a popular example. However, research in this fields typically focuses on the satisfiability of workflows and does not explicitly focus on SoD. As a result, we exclude work dealing with the WSP from this study. Research that explicitly focuses on SoD rules and also deals with the satisfiability of workflows is, however, included in our work[7].

[7]There are some borderline cases based on this definition. We have included these borderline papers for the sake of completeness.

Table 2. Categorization of relevant literature

Category	Publications	Total
SoD Specification		**39**
in Models		(24)
RBAC	[1, 2, 6, 20, 30, 32, 39–41, 46, 61]	11
ABAC	[9, 12]	2
Workflows	[21, 58, 70]	3
Others	[28, 51, 56, 63, 95, 97, 100, 103]	8
in Modelling Languages		(15)
Petri nets	[22, 44, 111, 112]	4
UML/OCL	[45, 74, 88, 90, 105]	5
BPMN	[108, 109]	2
Others	[43, 48, 85, 86]	4
SoD Enforcement		**26**
in Models		(12)
RBAC	[26, 36, 53, 54, 57, 93]	6
ABAC	[37, 38]	2
Workflows	[10, 14, 59, 84]	4
within a Specific Context	[16, 18, 23, 91, 110]	5
in Databases	[7, 8, 64, 81]	4
Implementation & Software Development	[3, 29, 62, 83, 89]	5
SoD Analysis		**17**
Algorithms concerning MER	[76–78, 87, 94]	5
Role Mining	[55, 82, 102]	3
Workflow Analysis	[11, 33, 52, 92, 96]	5
SoD Generation & Transformation	[4, 17, 42, 50]	4
High-level SoD & Management	[24, 25, 68, 71, 104, 107]	**6**

4.3 Results and Classification

We assigned the identified literature to the categories we previously introduced in section 4.1 of this work. The final results can be seen in Table 2. It lies in the nature of things, that there are some papers that would fit in more than one category. For example, a paper that introduces an approach to specify SoD in RBAC might also show a way to enforce SoD within the proposed specification. However, we decided to assign each paper to only one category, and in cases of conflict, the primary focus of the paper was carefully analysed and discussed. In the following, every publication considered in our work is shortly described following our introduced classification scheme. Additionally, we analyse and discuss the publications after each major section.

5 SOD SPECIFICATION

In general, SoD policies can be expressed verbally, e.g.: "All users working in the marketing department should not have access to the front office repository and the back office repository at the same time". However, it becomes necessary to formalize these rules in a language or syntax, that is compatible with existing models and frameworks. Therefore, existing models need to be extended in order to support the specification of SoD policies and rules. Research done in the area of specification can be divided into the subcategories "Specification in Models" and "Specification in Modelling Languages".

5.1 Specification in Models

In the following, we present the stream of literature that focuses on the specification of SoD within RBAC, ABAC, workflows, and some other models.

5.1.1 RBAC. SoD is extensively studied within the RBAC model [80][8]. In RBAC roles are set as intermediary between users and permission instead of assigning the permissions directly to the users. This reduces the number of assignments and facilitates the management of users and their permissions. RBAC also adds semantics to access control by giving roles meaningful names e.g. "Developer" for a role containing all permissions a developer needs. While RBAC0 is the core model and RBAC1 deals with role hierarchies the extensions RBAC2 and RBAC3 add Static and Dynamic SoD to the model. In RBAC2 the set $SSD \subseteq (2^{ROLES} \times N)$ is added as a collection of pairs (rs, n), where rs is a set of roles and n a natural number greater or equal 2. For each (rs, n) no user is allowed to have n or more roles assigned from rs. For example, $SSD = \{\{r1, r2\}, 2\}, \{\{r1, r3\}, 2\}$ would mean that the role r1 is mutual exclusive with roles r2 and r3. Static MER policies are specified by setting n=2. On a permission basis, SoD can be specified as $ssod(\{p_1, ..., p_n, k\})$, meaning that at least k users are required to have all the permissions $p_1, ..., p_n$ together. As this specification is often used, we will refer to them in the following as *default permission-based SoD constraints*. Regarding role-constraints the t-m SMER definition, which is very similar to the specification in RBAC2, is typically used: A t-m SMER constraint is expressed as $smer(\{r_1, ..., r_m, t\})$, meaning that no user is allowed to possess t or more of the defined m roles. Note that while t-m SMER constraints restrict the number of roles a user can possess from a role set, the default permission-based SoD constraints define the minimum number of users required to possess all permissions together from a permission set. The reason for this is, that SoD requirements can be defined on a permission basis (e.g. "at least two users are required to execute this process") while the enforcement is done via user-role assignments.

For Dynamic SoD, a specification is provided in RBAC3 by adding the set $DSD \subseteq (2^{ROLES} \times N)$ as a collection of pairs (rs, n). For each (rs, n) in DSD a user cannot activate n or more roles from rs within the same session i.e. at the same time. This allows the specification of Simple Dynamic SoD rules. The Dynamic SoD Specification on the foundation of exclusive roles is critically analysed by Habib et al. [32]. They argue that the concept of exclusive roles might be too strict on the one hand, because non-conflicting permissions contained in exclusive roles are treated like conflicting ones. On the other hand, it might be insecure as conflicting permissions in exclusive roles could be gained via other non-exclusive roles, if the roles are not defined properly. To address the challenges, they propose to specify SoD rules on the basis of conflicting permissions instead of conflicting roles. Then they divide the roles in two parts, the inner role containing all conflicting permissions, i.e. permission occurring in a SoD rule, and the outer role, containing all non-conflicting permissions. Therefore, permissions from the outer role can still be used, even if there would be a SoD violation caused by the activation of another mutual exclusive role. According to Habib et al. [32], another issue in the Dynamic SoD Specification is, that a user could activate one exclusive role in a session and then logoff and login again and activate another exclusive role within the new session. The implementation of an Object Based SoD property is suggested to address this issue. They also formally specify their proposed model and present a prototypical implementation.

Another way to define rules in RBAC is provided by Crampton [20]. In this scheme, each constraint is represented by a triplet (s, c, x) where s (= scope) is the set of entities affected by the constraint, c (= constraint) is the set of entities that must not be entirely possessed by s and x is the temporal context, which may be static (s), dynamic (d) or historic (h). A SoD constraint, that requires that no user is allowed to possess the roles r1, r2, and r3 can be expressed as

[8]See [80] for a formal definition.

$(U, \{r1, r2, r3\}, s)$ where U is the set of all users. Because the scope must not necessary be a set of users, it is also possible to define constraints for roles, permissions, transactions or objects in the RBAC-model.

For specifying SoD in the RBAC-model, Kuhn et al. [46] define a basic safety property, which has to be met in order to comply with SoD. The safety property states that no user is allowed to possess all roles necessary to perform a task, which is required by SoD to be executed by more than one user. Mossakowski et al. [61] claim that the plain RBAC-model is not capable of specifying Object Based Dynamic SoD (see [62]) and propose an extension in which the execution history, especially the operations on objects, is kept in order to specify and enforce temporal constraints.

A more exhaustive work in a similar direction is done by Joshi et al. [39, 40]. They define the Generalized Temporal Role Based Access Control Model (GTRBAC) [39] as an extension of the RBAC-model in order to allow the specification of temporal constraints i.e. Dynamic SoD. In their model the time points within a given time interval are considered as equal. For example, "SoD $\wedge$ active$(u, r1, t) \rightarrow \neg$active$(u, r2, t) \mid u \in U, r1, r2 \in R$" means that no user from U is allowed to have more than one role from R active within a defined time interval. This is the basic definition of Simple Dynamic SoD with the roles of R being in a mutual exclusive set. In their work they show that their model is capable of specifying the complete list of SoD constraints presented by Simon and Zurko [89].

In order to study SoD properties, Ahn et al. [1] propose the Role-based Separation of Duty Language 1999 (RSL99) [1], which also takes the basic elements from the RBAC-model [80]. RSL99 utilizes the two set operations $OE(X)$ returning one object from the set X (but always the same one) and $AE(X)$ returning all other objects from X except $OE(X)$. For a set of users U and a collection of conflicting role sets CR the Static SoD property could be specified as $|roles(OE(U)) \cap OE(CR)| \leq 1$ meaning that the intersection between the roles of an element from the user set U and the roles from an element of the conflicting role sets CR must not contain more than one element. In other words, a user cannot have more than one role of each conflicting role set. Their model is flexible in the way of expressing SoD properties. The Role-Based Constraints Language 2000 [2], the successor to RSL99, extends RSL99 to support obligation constraints, which are beyond the context of this paper.

Kallel et al. [41] apply the formal specification language TemporalZ [75] on RBAC in order to specify SoD. A TemporalZ expression consists of domain specific predicates and nested other expressions using the temporal operators: $\Box$ (always), $\Diamond$ (eventually), $\bigcirc$ (next), $\boxminus$ (up till now), $\boxtimes$ (previously), $\ominus$ (at last), $\mathscr{U}$ (until), and $\mathscr{S}$ (since). Hence, Kallel et al. [41] define RBAC specific TemporalZ predicates like $include(u, r)$ or $active(u, r)$ which become true when role r is included or currently active for user u. Using the temporal operators, it is possible to specify SoD rules e.g. a Simple Dynamic SoD rule can be expressed as $\forall\, u : USER \bullet \Box(\neg(active(u, r1) \wedge active(u, r2)))$. The approach also supports Static SoD, Object Based SoD and Operational Based SoD using additionally defined predicates and also allows the specification of more fine-grained SoD properties. Further a mapping to the aspect-oriented language ALPHA [69] is provided to implement security monitors in order to realize the practical evaluation and enforcement of the specified SoD rules.

5.1.2 ABAC. In ABAC [35] authorization rules, based on attributes, define the actions a user can perform on an object. The authorization rules consist of the users' attributes, the objects' attributes as well as environmental attributes like the time or the location. Formally, an ABAC policy usually contains a set of users U, a set of objects O, with respective sets of attributes UA, OA and a set of authorization rules P. Additionally a set of environmental attributes EA may be included. The authorization rules are defined in the form (uc, oc, ec, op) with uc, oc, ec as sets of users, object and environmental conditions and op as allowed operations. The conditions have the form of $attribute = x$ where x can be

a concrete attribute value or any. For example, an authorization policy allowing developers to write on source code objects could look like $p1 = (\{function = Developer\}, \{document\_type = source\_code\}, \{time = any\}, \{write\})$.

Bijon et al. [12] propose the attribute-based constraint specification language (ABCL) for SoD and constraint specification in ABAC. ABCL reuses some features of RSL99 and RCL2000, for example the one element operation OE, and also allows the definition of constraints with similar expressions. To express SoD, the authors recommend defining a set attribute for users (we call this attribute "functions" in the following) to keep track of the roles or functions assigned to a certain user. Conflicting roles or functions are defined in an additional set "ConflictingFunctions". Hence, Static SoD could be expressed as $|OE(ConflictingFunctions).attval \cup role(OE(U))| \leq OE(ConflictingFunctions).limit$.

This means that for each one element of "ConflictingFunctions" no user with more than the specified limit of functions in common must exist. Note that the function *attval* returns the attribute values of a "ConflictingFunctions" element (e.g. "Developer", "Accountant"). Additionally, Dynamic SoD constraints can be specified with ABCL, a full specification of the syntax in Backus Normal Form can be found in [12]. Another way for SoD Specification within the ABAC model is proposed and used by [9, 37, 38]. Similar to some RBAC expression schemes, the rules are expressed as a set of action-operation pairs and a cardinality defining how often a pair is allowed to be used. For example the rule $r_1 =< \{(create, loan\_application), (approve, loan\_application), 2\} >$ defines that a loan application must not be created and approved by the same user. It might be noted that this is a rather straight-forward but practical expression scheme used by works which focus on SoD Enforcement in ABAC [37, 38].

5.1.3 Workflows. Workflows are one of the typical use cases for SoD, especially for Dynamic SoD. Therefore, work has been done to specify, enforce and analyse SoD in workflow environments. In general, when applying SoD on workflows an underlying access model (e.g. RBAC) is used to determine whether a user or role is allowed to execute a task of the workflow.

Crampton et al. [21] propose a SoD Specification for workflows, in which this dependency is generalized. Therefore, they define the constraint workflow authorization scheme as a triplet (T, A, C), where T is a set of tasks, A is a set of authorizations, and C is a set of entailment constraints. The underlying access model is generalized by A as follows: $A \subseteq T \times U$, where U is a set of users. A user u can perform a task t iff $(t, u) \in A$. Therefore, A can be derived from any access control model and the constraints can be defined separately in C. An entailment constraint has the form $(D, (t, t'), p)$. D defines the domain i.e. the users for which the constraint is valid, t and t' are two tasks, where t' is a subsequent task of t. Finally, p denotes a binary relation between users, e.g. equals, non-equal or any other arbitrarily complex operator. A simple mutual exclusive SoD rule between two tasks (e.g. t1, t2) can be expressed as $(U, (t1, t2), \neq)$. Binding of Duty[9] (BoD), i.e. t1 and t2 must be executed by the same user, can be specified by $(U, (t1, t2), =)$. It might be noted, that the Domain can be specified arbitrarily, for example "all users in a specific department" or "all users working as developers" and also the binary operator may be enriched with specific context like "is supervisor of". Beside the specification, the work of Crampton et al. [21] also presents an algorithmic approach for SoD Enforcement by a reference monitor and handles workflow satisfiability, which is beyond the scope of this paper.

To specify SoD in the Web Services Business Process Execution Language (WS-BPEL) Bertino and Crampton [70] propose the Business Process Constraint Language (PBCL) as an extension. WS-BPEL is a language based on XML for the specification of business processes. It does not support access control or authorization constraints out of the box [70]. Activities of a workflow can be arranged in a specific order, for example the element <sequence> puts activities in a sequential order and there are also elements for parallel or exclusive execution. In a first step, they extend WS-BPEL with

[9] A BoD constraint specifies that some tasks must be executed by the same role or user

RBAC (RBAC-WS-BPEL) by mapping the RBAC permissions to the activities of WS-BPEL in a way that a permission allows a user to perform a specified action on an activity. For the specification of the RBAC components they utilize the RBAC XACML policy language. In a second step, they introduce the XML-based Business Process Constraint Language (BPCL), which is capable of specifying SoD policies. In addition to SoD, BPCL supports similar constraints like the specification of BoD or the consideration of role hierarchies in a way that a senior or junior role must or must not execute a consecutive task.

Work in a similar direction is done by Mendling et al. [58] on a high level by discussing strategies and potential issues when specifying SoD in the BPEL extension BPEL4People, which focuses on modelling human interactions.

5.1.4 Other research on SoD Specification in models. Additionally, there are other works focusing on the specification of SoD in models briefly mentioned below. Foley [28] focuses on specifying and enforcing SoD within information flows. He develops the notion of information flow policies to define SoD constraints. Tidswell and Jaeger [97] propose an approach to define SoD in the dynamically typed access control model (DTAC) [98]. Takabi et al. [95] use fuzzy logic for the specification of SoD in RBAC.

An algebra for specifying high level policies is developed and explored by Li and Wang [51]. The algebra is capable for defining SoD rules and provides the operators $\neg, +, \sqcap, \sqcup, \odot, \otimes$ which are briefly explained by examples in the following: With $\sqcap$ and $\sqcup$ conjunctive and disjunctive combinations of roles or sets of users can be created, e.g. ("Developer" $\sqcap$ "internal") requires a user to be an internal developer and ("Developer" $\sqcup$ "Manager") requires a user to be a developer or a manager. $\neg$ creates a blacklist e.g. ("Developer" $\neg$ {"Alice", "Bob"}) requires a developer which is not Alice or Bob. The term ("Developer" $\odot$ "Manager") requires a user who is manager and a user who is developer or a user who is both. In contrast the term (("Developer" $\otimes$ "Manager") $\otimes$ All) requires three user, one who is developer, one who is manager and any other user (the keyword All refers to the set of all users in the algebra). Developer+ requires a set of one or more developers. Therefore, a typical SoD rule requiring an accountant and an approver for a task, could be ("Developer" $\otimes$ "Approver"), however the algebra is capable of defining far more complex expressions and rules. A detailed formal definition and exploration of the algebra can be found in [51].

Nguyen et al. [63] extend the provenance-based access model (PBAC) in order to support Static and Dynamic SoD including Object Based SoD, Operational Based SoD and History Based SoD. Lu et al. [56] develop an approach for specification and enforcement of Static SoD in the usage control model, a model that covers, beside of authorizations, the concepts of obligations, conditions, continuity and mutability [73].

Work has been done by Veloudis and Nissanke [103] to integrate Static SoD in multi-level security mandatory access control (MAC) models, such as the Bell-LaPadula (BLP)-model. In the BLP model, objects and users are assigned to security levels, e.g. "confidential", "secret" and "top secret". Users can read objects that are lower or equal to their security level and can write to objects that are higher or equal to their security level. In our example a person classified as "secret" can write articles that are "secret" or "top-secret", and can read articles that are "secret" or "confidential", ensuring that no information is passed from a higher to a lower security level[10].

In order to specify Static SoD policies within the BLP-model Veloudis and Nissanke [103] utilize features of the RBAC-model. Therefore, permissions in RBAC are assigned to a capacity based on the object they refer to, e.g. if a permission grants write access to a document classified as "secret", the capacity of the permission is "secret". Additionally, a permission hierarchy is defined in way that it applies to the BLP-model, which for example means that a write

[10]In [103] other operations than "read" and "write" are possible, but for the sake of simplicity we stick with these two.

permission for a document classified as "secret" inherits all permissions from security levels equal or higher than "secret".

Each role is assigned to a security level and only permissions with the corresponding capacity may be assigned to that role. To avoid redundant permission-role assignments, only zero or one write and only zero or one read permissions can be assigned to each role, resulting in roles with only two permissions with all other permissions inherited indirectly via the permission hierarchy. Individual roles can be created, by explicitly excluding inherited permissions from roles. For example, consider permission p1 and p2 as read permissions on documents with the capacity "confidential" and p3 as read permission with the capacity "secret". A role r1 containing p3 would implicitly include p1 and p2, but it is possible to explicit exclude p1 that $r1 = \{p2, p3\}$. On this foundation, Static SoD based on mutual exclusive permissions is applied by removing conflicting inherited permissions from the defined roles and by calculating the resulting exclusive roles. Consider a mutual exclusive permission pair $sod = \{p2, p3\}$. The role r1 would be modified to $r1 = \{p3\}$ in order to avoid an inherent SoD conflict and r1 would become mutual exclusive with all roles containing p2.

A more generic approach to specify Dynamic SoD is provided by Ultra and Pancho-Festin [100] by proposing the concept of validation policy as an extension for any access model. Instead of defining mutual exclusive roles or permissions, they suggest adding validation policies to the permissions of the access model. The combination of a permission and a validation policy is called a task. Each validation policy consists of several validation entries and each validation entry contains a function to determine the set of users allowed to perform the validation and the amount of validations necessary to be satisfied.

For a validation policy it is also possible to consist of zero validation entries. In this case the permission can be executed immediately. If this is not the case, a validation request is created on permission activation and the users specified by the validation entries' function are ask for their approval. Because the allowed users for a validation entry are dynamically determined by function, Dynamic SoD properties are reached. For example, a typical way to define the allowed users function of a validation entry is to select all users assigned to the concerning permission instead of the user who wants to activate it.

5.1.5 Discussion of Research on SoD Specification in Models. The goal in specifying SoD for models is to provide means in such a way that SoD can be expressed and integrated in the respective models. There are different approaches to achieve this: In many publications addressing access models (RBAC, ABAC, etc.), a key objective is to add a broad spectrum of the various SoD properties. These approaches extend the access models by adding formalism for specifying SoD rules, as well as definitions that ensure these rules are adhered to. Table 3 provides an overview of these publications and indicates which SoD properties are targeted. We selected the commonly used SoD properties proposed by Simon and Zurko [89]. The table includes only work that address the topic of enhancing models with SoD properties. Publications by the same authors that deal with the same artifact and build upon one another have been consolidated into a single line.

When analysing Table 3, it becomes apparent, that every research work supports Static and most works support Simple Dynamic SoD properties. Static SoD properties are easy to implement and considered strict but useful in practice. The practical value of Simple Dynamic SoD properties is debatable, as a user can activate conflicting assignments at different times (cfg. [89]).

Object Based and Operational Based SoD properties are supported by roughly half of the publications each and can be useful depending on the application. History Based SoD properties are very flexible and considered to be meaningful

in practice [20, 89]. However, publications that support History Based SoD properties require an extension that captures and stores the execution history of actions and accesses (e.g. [20, 41, 61]).

Finally, most publications refer to a specific access model (mostly RBAC). Only two models [51, 100] offer a generic approach regarding the underlying access model.

In order to enhance workflows to support SoD policies, access models are used as underlying mechanism for access control. Two works specify SoD in workflows by extending BPEL: WS-BPEL [70] and BPEL4People [58]. While WS-BPEL is used to orchestrate web services, BPEL4People focuses on modelling human interactions. Both works use the RBAC model as underlying access model. However, [70] provides a concrete specification of SoD in WS-BPEL by introducing RBAC-WS-BPEL, while [58] discusses the issue only on a high level. The work of Crampton et al. [21] allows the use of any access models and deals with entailment constraints (e.g. SoD, Binding of Duty) in workflows. It also addresses the topic of workflow satisfiability.

Pub.	Model	Static	Simple Dynamic	Object Based	Operational Based	History Based
[32]	RBAC	x	x	x	-	-
[20]	RBAC	x	x	x	-	x
[61]	RBAC	x	x	x	x	x
[41]	RBAC	x	x	x	x	x[11]
[1, 2]	RBAC	x	x	-	-	-
[95]	RBAC	x	x	-	-	-
[12]	ABAC	x	x	-	-	-
[9, 37, 38]	ABAC	x	-	-	-	-
[39, 40]	GTRBAC	x	x	x	x	x
[97]	DTAC	x	x	x	x	x
[63]	PBAC	x	x	x	x	x
[56]	UCM[12]	x	-	-	-	-
[103]	BLP	x	-	-	-	-
[100]	Any	x	x	x	x	x
[51]	RBAC/ Any	x	x	x	x	x

Table 3. Research on SoD Specification in models that addresses SoD properties

5.2 Specification in Modelling Languages

Extending modelling languages to support SoD provides, among other advantages, a way to enrich SoD rules with semantics and visualisations. When extending the modelling languages, the RBAC model is usually utilized as the underlying access model.

5.2.1 Petri nets. PNs are used to specify and analyse SoD policies in the workflow context. So-called Petri net workflows (PNW) are used to specify the tasks and the control flow of workflows within PNs. Therefore, PNWs are particularly interesting to study Dynamic SoD. A PN is defined as $N = (P, T, F)$ where P is a set of places, T is a set of transitions and $F = (P \times T) \cup (T \times P)$ is a set of directed arches connecting the places with the transitions. Tokens can be set on places, and all transitions which can be reached from tokens with places are called active transitions. Active transitions can "fire" and thereby move the connected tokens from one place to another according to the directions of the arches.

[11] History Based SoD is called Operational Object-Based SoD in [41].
[12] Usage Control Model [73].

The number of tokens at a place p can be acquired by the function M i.e. $M(p1) = 1$ means that there is one token on place p1. First work in SoD Specification within PNWs has been done by Knorr and Weidler [44] by mapping the tasks of a workflow as the transitions of the PNs and defining an underlying role model with R as a set of roles and S as a set of subjects. SoD rules are specified in the form of $(s1, t1) \rightarrow \neg(s2, t2) \mid s1, s2 \in S, t1, t2 \in T$. This means that if subject s1 has activated the transition t1 (i.e. task) then subject s2 is not allowed to activate transition t2 for this PN instance anymore, allowing the specification of Dynamic SoD. They also provide a practical implementation in Prolog for analysing workflows and SoD rules. This means that the enforcement of rules is not directly modelled or implemented in the PNW, but in the Prolog program. For example, for analysis purposes, the Prolog program can calculate all possible execution chains that do not violate any SoD rules.

Carvalho and Bandiera-Paiva [22] propose a model based on Colored PNs that aims to represent a RBAC system and SoD rules. Rather than simulating workflows, this model is intended to simulate the RBAC system itself (i.e. the users, role activation etc.). Therefore, colored tokens are used to represent RBAC entities (e.g. roles, users etc.). To demonstrate its applicability, the authors implement the model in a Colored Petri Net simulator. Although the publication originated in the healthcare sector, the modelled RBAC system should be generally applicable.

Yang and Hu [111] present a specification scheme with tasks represented by the places in the PNW (instead of transitions). They specify an underlying RBAC model within their PNs in a way that the users, roles and the permissions of RBAC are represented as places and their hierarchical order is structured using transitions and arcs. If a user activates a permission the corresponding place gets a token, if the user stops using that permission, the token is removed. In the following we consider p1 and p2 to be places and P1 and P2 to be their corresponding permissions. SoDs can be expressed as linear inequalities in the form of $M(p1) + M(p2) \leq 1$ meaning that the number of tokens at p1 and p2 must be equal or less than one and therefore representing a Dynamic SoD policy with mutual exclusive permissions P1 and P2. To enforce this policy at runtime, they propose the concept of critical places, which are defined as the places capable of leading to a SoD violation (in our case p1 and p2 would both be critical places). If a critical place has a token, the transitions leading to the other critical place defined in a SoD rule are deactivated and cannot fire. If a user has a token on p1 and therefore activated the corresponding permission P1 she is not able to fire any transition to p2 and therefore cannot activate permission P2. Consequently, in a subsequent work [112], Yang and Hu address the enforcement and conflict resolution for SoD and BoD conflicts in petri nets-based RBAC models.

5.2.2 UML/OCL. Work has been done to specify SoD in the Unified Modelling Language (UML) [45, 74, 88, 90, 105]. Therefore, the RBAC model can be expressed as a class model in UML. Because UML does not support constraint specifications itself, the Object Constraint Language (OCL) is utilized to specify SoD rules. Sohr et al. [90] additionally propose OCL expressions to specify Dynamic and History Based SoD constraints and provide a tool-supported validation by using the UML Specification Environment tool. Ray et al. [74] argue that the UML/OCL modelling approach might be hard to use because the OCL constraints are difficult to understand. They propose the use of UML Object Diagram Templates, which are able to visualize OCL constraint violations, in order to address this issue. Note that there is no way provided to translate Object Diagram Templates into OCL. As a result, the approach aims to support UML modelling with respect to existing SoD constraints already specified in OCL.

5.2.3 BPMN. The BPMN is a well-known means for the visual specification and modelling of workflows. To specify SoD in BPMN, Wolter and Schaad [108] argue that most approaches rely on separate SoD Specification and workflow modelling environments (e.g. UML/OCL) and propose an integrated solution in which the SoD rules can be directly specified within the BPMN diagram. In a first step, they propose a notion for constraints with the form of $c = (T_c, n_u, m_{th})$

with T_c as a subset of the tasks of the BPMN diagram and $n_u, m_{th} \in N$. Within the execution of one workflow instance the parameter n_u defines the minimum number of users that must be involved in the execution of the tasks in T_c, and m_{th} defines the maximum number of tasks a single user is allowed to execute in T_c. A simple example is the constraint $c = (\{t_1, t_2\}, 2, 1)$, which means that t_1 and t_2 cannot be executed by the same person in one workflow instance, because the maximum number of tasks a user is allowed to execute within T_c is 1. In order to integrate this scheme directly into the BPMN diagram, an authorization constraint, which inherits from the textual BPMN notion, is added as new artifact. The authorization constraint has the form (n_u, m_{th}) and is assigned to groups, lanes or directly to tasks in order to specify the tasks of T_c. In a consecutive work [109], a mapping for the constraints from the BPMN notion to XACML access control policies has been proposed. Wolter et al. [109] argue that the specification with their BPMN notion is more intuitive, while XACML is widely supported and therefore can better be enforced.

5.2.4 Other research on SoD Specification in modelling languages. The MASoD (Modelling and Analysis of SoD-rules) tool is specially designed for the visual modelling and analysis of SoD in workflow environments [43]. Like other approaches to model workflows and SoD, the tool supports a role-model, a workflow-model, and a SoD-model. An Event-driven Process Chain (EPC) is proposed as process-model in which the functions are used to model the tasks of the workflow. The roles of the role-model authorize a user to execute a specific function i.e. to perform a specific task. To model SoD rules the Simple SoD Language (SSoDL) and its graphical representation are introduced. Rules have the form $(u_1, t_1) \nrightarrow (u_2, t_2)$ meaning that if user u_1 has executed task t_1, the user u_2 is not allowed to execute task t_2. To create meaningful rules, a ?-wildcard is introduced in a way that $(?, t_1) \nrightarrow (?, t_2)$ means that a single user is not allowed to execute both, t_1 and t_2, in the same workflow instance. Compared to other approaches, MASoD offers little that is new but is one of a few approaches that focus on graphical modelling.

For the visual modelling of RBAC and SoD in the Web Ontology Language (OWL), Kwon and Moon [48] propose the use of OWL DL [15] expressions and OWL DL reasoners like RACER or JENA to detect conflicts. Specifying RBAC and SoD using the alloy specification and modelling language has been studied by Schaad et al. [85, 86]. They provide means to specify Dynamic and Static SoD and pay especially attention to the delegation of obligations.

5.2.5 Discussion of Research on SoD Specification in Modelling Languages. The publications dealing with PNs differ significantly in their objectives. While most publications [44, 111, 112] focus on workflows and SoD rules, the work of Carvalho and Bandiera-Paiva [22] aims to model RBAC at runtime using Colored PNs (supporting the Static SoD property).

In [44], only the workflow is modelled with PNW, and the enforcement is achieved through a Prolog program. The work is focused on analysing workflows. The work of Yang et al. [111, 112] models both, permissions and users within the PN and implements SoD using linear inequalities. This is a structurally integrated approach, as both workflow and permissions are modelled within PNs. However, users, permissions, and other entities are modelled as places. In practice, organizations can be very large with thousands of users and permissions, which will result in very large and complex PNs.

The publications regarding UML/ OCL deal with the question of how RBAC states and SoD can be modelled. Therefore, OCL is used to specify SoD rules. They can be compared with respect to the supported SoD properties. This comparison can be seen in Table 4. While most work supports Static and Simple Dynamic SoD properties, only the work of Sohr et al. [90] supports all SoD properties. It should also be noted that the work of Koch and Parisi-Presicce [45] does not focus on SoD properties but provides a comparison between different modelling options (UML, Alloy, Graph Transformation).

Since all approaches to UML/ OCL do not consider workflows, the work of Wolter et al. [108] is of importance. By extending the BPMN notation to support SoD, it is possible to model the SoD restrictions directly into the workflow. Their consecutive work [109] allow the automated derivation of XACML policies that can be passed directly to the corresponding systems for enforcement.

Pub.	Static	Simple Dynamic	Object Based	Operational Based	History Based
[45]	x	-	-	-	-
[88]	x	x	-	-	-
[105]	x	x	-	-	-
[74]	x	x	-	-	-
[90]	x	x	x	x	x

Table 4. Research on SoD Specification with UML/ OCL and supported SoD properties

6 SOD ENFORCEMENT

The enforcement of SoD deals with the challenge of detecting or avoiding SoD violations. This can be analysed within the context of a specific model e.g. RBAC, ABAC or workflows. Some literature also deals with enforcement under various circumstances, like decentralised systems. Especially some early work explores the enforcement within databases and finally, some literature focuses on implementing SoD and integrating it in the software development process. The following section provides a summary of the literature on SoD Enforcement. A discussion of this literature is presented at the end of the section.

6.1 Enforcement in Model

6.1.1 RBAC. Dynamic SoD in RBAC is enforced during role activation, which is studied in the work of Lui et al. [57]. They argue that keeping track of the activated roles is crucial for enforcing SoD at run-time. Therefore, an autonomous management approach, in which each application maintains its own list of activated roles, is insufficient. A centralized management approach, with one central instance maintaining the activated roles is more feasible, however, this approach creates a bottle neck and a single point of failure as well as lacks scalability. Therefore, Lui et al. [57] suggest a decentralized management approach, a generalization of the centralized approach, in which multiple role activation services in different domains (e.g. organizational units or locations) keep track and handle the role activation as well as synchronize the list of activated roles among each other. The domains are organized in a hierarchical order and each domain can have its own SoD rules specified. When a user wants to activate a role in a specific domain, the SoD rules of that domain as well as the rules of all ancestor domains must comply. This allows coarse grained rules to be specified at higher levels of the domain hierarchy and more fine-grained rules to be specified at the lower levels, where the domain-administrators might have the respective know-how.

The reliability of the SoD specification of the ANSI/ INCITS RBAC standard [6] is analyzed in the work of [26]. They claim that one vulnerability of the specification (Static and Dynamic SoD) is the existence of redundant roles (see also [32]). If roles r1 and r2 are mutually exclusive, it might still be possible to gain all permissions of r1 and r2 through other roles. In their work [26], they propose several ways to update the standard to avoid such a constellation.

Song et al. [93] examine SoD in context of the administrative role graph model, which focuses on role hierarchies and domains. They provide several algorithms to alter the role graph in respect of SoD e.g. by adding or removing roles

or domains. Jeon and Yoo [36] study the implication of delegations within the RBAC model on SoD Enforcement. They use cellular automata in order to detect conflicts when delegating roles from one user to another.

In their work Liu et al. [53, 54] propose an enforcement mechanism for static mutually exclusive roles constraints in the Interoperable RBAC (IRBAC) model. The IRBAC model is capable of translating roles from one RBAC domain to roles in another RBAC domain by using associations. One challenge in the IRBAC model is to enforce SoD but still preserve privacy i.e. the RBAC domains are not sharing information about their roles and role assignments with each other. Their approach uses a combination of encryption schemes and matrix multiplication of role-hierarchy and role-assignment matrices to detect SoD violations and therefore enable SoD Enforcement.

6.1.2 ABAC. In order to enforce SoD within ABAC, Jha et al. [37, 38] define the ABAC SoD verification problem. It answers the question whether an ABAC state, i.e. the ABAC policies, the objects, the users and their respective current attributes, satisfy a set of SoD policies. The SoD policies are expressed as action-operation pairs and a cardinality defining how often a pair is allowed to be used (see section 5.1.2). They show that the original ABAC SoD verification problem is intractable and provide a brute force algorithm for solving it. In order to find a more performant solution, they propose the idea of generating mutual exclusive policies from the SoD policies and define the mutual exclusive policy verification problem. Similar to RBAC, mutual exclusive policies are expressed as $(\{p1, p2\}, 2)$ meaning that in an ABAC state, policy 1 and policy 2 must not apply to more than one user. In [37] and [38] algorithms for transforming, generating and verifying mutual exclusive policies are provided and it is shown that the mutual exclusive policy verification problem can be solved efficiently.

6.1.3 Workflows. Botha and Eloff [14] focus on the enforcement of SoD within role-based workflow environments. In their model, the workflow tasks are assigned to users and the roles enable the users to perform these tasks by granting the necessary permissions. In order to allow more flexibility when implementing SoD policies, they present the "conflicting entities administration paradigm", allowing SoD rules to be expressed as conflicting roles, conflicting permissions, conflicting users (e.g. family members) or conflicting tasks. They argue that Static SoD can be enforced on an administrative level by avoiding conflicting entity assignments.

To enforce Simple Dynamic SoD, especially in role-hierarchies, the concept of WSessions is proposed. When a user performs a task in a workflow instance, a WSession is created, which selects and activates the most junior-role of the user necessary to execute that task. This role must not directly be assigned to the user but can also be within the role-hierarchy connecting the user with the necessary permissions. This enables a senior-role to contain (Dynamic) SoD violations, but when a user activates the role, she acts through a less powerful junior-role which enables more fine-grained Dynamic SoD Enforcement. They also present a practical implementation of their approach.

Schaad et al. [84] propose a model checking approach (i.e. examination of a workflow for potential SoD violations) for delegation and revocation functionalities in workflow-based ERP systems, i.e., the transfer of tasks and permissions to another user or the revocation of such assignments. In their approach, the access control policies, the workflow, as well as the delegation and revocation functionalities are modelled as a finite state machine and the SoD rules are specified in linear temporal logic. Several safety properties which are categorized in SoD-based safety, task-based safety, and delegation and revocation-based safety are defined. Based on this specification, an analysis is conducted to determine necessary steps and checks required to maintain the safety properties in respect to different SoD properties (e.g. Static SoD, Simple Dynamic SoD, Object Based SoD etc.). Another model checking approach observing workflows with uncertain information (e.g. how long a task takes) is presented in [59].

Basin et al. [10] extend the previously mentioned algebra for specifying separation of duty policies and rules by Li and Wang [51] in order to enable enforcement mechanisms especially during workflow execution. Therefore, a way to map terms of the algebra to processes (in particular to the process algebra CSP [34]) is proposed.

6.2 Enforcement in a specific Context

The enforcement of Static and Simple Dynamic SoD in RBAC within the context of non-centralised and non-homogeneous environments is studied in the work of Chadwick et al. [16]. In their environment, there is no central authority that knows the users full set of roles or assignment history, and processes may span over multiple user sessions at different times. For example, a user may be assigned to role r1 at a specific time point and a long time later, after r1 has already been revoked, the user might be assigned to role r2, which is in Static conflict with r1. However, as the user never had r1 and r2 at the same time point, the SoD rule will never be violated. A practical example could be, that role r2 is an auditor role and the user could now audit her own actions done via role r1. Therefore, they propose the multi-session SoD policy (MSoD), which additionally defines a so-called business context. A business context is a set of business processes in which a SoD policy must be enforced. The authors recommend a hierarchical order of business contexts with the "universal context", containing all business processes, on the top of the hierarchy.

More fine-grained business contexts can include temporal or spatial information like "Branch = York, Period = 2024", containing all business processes from the York office in 2024. The enforcement of MSoD policies (in a non-centralised and non-homogeneous environment) needs to be done at the access control decision making stage, as the MSoD policies are tied to several business processes (via business context) and may span over several sessions. Therefore, the ISO Access Control Framework LINK[13] is enhanced with the current business context (in order to determine which MSoD policies apply) and a so-called Retained Access control Decision Information component (ADI). The ADI manages the information about previous access control decisions. With the additional information of the ADI and the current business context, MSoD policies can be enforced when access is requested. Their work [16] also contains an XML specification for MSoD policies as well as a prototypical implementation of their enforcement approach within the Privilege Management Infrastructure PERMIS.

For enforcing SoD and other policies in the context of web services, Sohr et al. [91] propose an authorization framework with a generic "Authorization Engine" as core, that utilizes the USE system[14] to support UML and OCL. The framework uses a "Policy Decision Handler" as an interceptor and policy enforcement point i.e. it acts as a web service for the clients and as a client for the "Authorization Engine" and the actual web services. When a client wants to execute a web service, it sends a request to the "Policy Decision Handler", which handles the authorization with the "Authorization Engine", and then forwards the request to the actual web service. Their proposed implementation of the "Authorization Engine" with USE supports RBAC, Static SoD, Simple Dynamic SoD, and History Based SoD.

Additional notable work on the enforcement of SoD in a specific context include the research conducted by Chen et al. [18], which focuses on SoD Enforcement in a spatial, location-based context. Deng et al. [23] examine the enforcement of SoD within ad-hoc collaborations, and Yang and Hu [110] propose algorithms to avoid SoD violations in multi-domain environments.

[13] The ISO Access Control Framework consists, among other things, of a policy enforcement point, which handles a decision request, and a policy decision point, responsible for deciding if access is granted.
[14] The USE system is a JAVA-based validation tool for UML models and OCL based constraints.

6.3 Enforcement in Databases

Early work to enforce the SoD properties (from Static up to History Based) with focus on databases and a finite amount of storage space has been done by Sandhu [81] by proposing transaction control expressions (TCE) for SoD. They distinguish between semantic transient and semantic persistent objects. For example, when opening a bank account, the bank account itself is a persistent object and the request form to open the account is a transient object. The main difference is, that a persistent object can develop an arbitrarily history, while a transient object has a finite history e.g. an account opening form is filled out, reviewed, and finally the account is created. The transient objects are used to modify the persistent ones e.g. the opening form creates an account, a change request modifies an account etc. At first, a TCE is a list of the possible future steps of a transient object and the users or roles permitted to perform these steps. Later, the user who actually performed the step is stored. For example, consider the following TCE for a form to create a bank account and a SoD policy which requires each step to be carried out by a different clerk:

fill out form • clerk; review form • clerk; create account • clerk

After the form has been filled out by clerk Jerry, the TCE would look like this:

fill out form • Jerry; review form • clerk; create account • clerk

It is now possible to check whether the SoD rule is violated when performing the next steps, and as the form is a transient object with a finite number of steps, the required storage space in the database can be kept small. An implementation of this approach using a typed access matrix is proposed in [7]. Another approach is presented by Nyanchama and Osborn [64], by proposing an object-oriented database model in combination with roles. Similarly to the TCE approach [7, 81] they keep track of an objects history by storing the audit-trail information directly at the database object itself to enable a performant enforcement of Dynamic SoD.

Another challenge comes up when storing data in encrypted cloud databases. In this case, the enforcement of policies, which requires the storage of access information, also has to be done in a secure way to avoid leaking information during the enforcement process [8]. In order to achieve this, Asghar et al. [8] propose an architecture with an outsourced enforcement module that is capable of enforcing policies in an encrypted manner. One key idea is, that every user has her own encryption key, and that the matching of attributes or other access data is done without comparing plain text.

6.4 Implementations and Software Development

A prototypical implementation of an authorization engine for distributed applications supporting SoD is Adage [89]. Adage enables the design and enforcement of SoD by providing a visual policy builder and a textual authorization language (called AL). It supports Static, Simple Dynamic and History Based SoD.

Another implementation for Dynamic (Simple Dynamic, Object Based and Operational Based) and Static SoD using Prolog has been provided by Schaad [83]. They demonstrate the straight-forward applicability of Prolog to simulate the RBAC model and SoD constraints using Prologs facts and rules. For example, Static SoD can be implemented as follows:

```
staticsod(User, Role1, Role2):- holds(User, Role1), mutex(Role1, Role2), holds(User, Role2).
```

Their approach additionally considers delegations within the RBAC model, i.e., a scenario in which a user temporary delegates her roles to another user (maybe because of vacation or illness).

A programming language based approach for SoD in java-like programming languages is proposed in [29]. By using the built-in annotation system of java, classes can be assigned to roles and the access to methods of instances of other

classes, with which they are mutually exclusive to, can be limited by the class loader. This is useful when loading additional content during run-time (e.g. extensions) or when the development process includes a third party.

SoD as a software requirement during the planning and specification phases of software development is studied in the work of Alghathbar [3]. AuthUML, a framework for analysing access control requirements in software development with use cases, is extended. The original AuthUML offers a process with 3 phases ("Initial/ First" phase (1), "On use case level" phase (2), "On operation level" phase(3)) for detecting conflicts and inconsistencies within the specification of access control requirements. Those phases include rules which represent the policies defined by the requirement engineers. $cando_{UC}(Manager, +"sign\ a\ check")$ e.g. means that the role "Manager" is allowed to access the use case "sign a check"[15]. Alghathbar [3] enhances AuthUML with a fourth "Validation SoD" phase, which validates SoD policies in four steps: at the requirement stage, when users are assigned to roles, when the roles are activated and when the users try to execute an operation. Additionally, AuthUML is enhanced with new predicates and rules in order to define and evaluate conflicts. For a simple example $alert_{sod} \leftarrow User\_Role(X_u, X_r), User\_Role(X_u, X_r'), conflictingRole(X_r, X_r')$ would alert the requirement engineer if a user is assigned to two conflicting roles. Furthermore, more complex and Dynamic SoD properties like task or operational-based SoD can be expressed with the extension.

6.5 Discussion of Research on SoD Enforcement

In the following we discuss the publications that deal with the enforcement in access models (i.e. RBAC and ABAC) that do not focus on a specific context like the work summarized in section 6.2. When examining the supported SoD properties (see Table 6), it may initially seem surprising that studies specifically focused on SoD Enforcement only address the Static and Dynamic SoD properties. However, this can be explained by the fact that the works have different objectives than fulfilling SoD properties. For instance, many studies [36, 50, 54, 93] aim to maintain SoD during changes (e.g., delegations) or, in the case of the IRBAC model, to preserve privacy while enforcing SoD. The work of Lui et al. [57], on the other hand, explicitly addresses dynamic SoDs. In ABAC, initial work has been done towards Static SoD properties [37, 38]. Exploring Simple Dynamic, Object Based, Operational Based, and especially History Based SoD properties remains a subject for future research.

Pub.	Static	Simple Dynamic	Object Based	Operational Based	History Based
[57]	x	x	-	-	-
[26]	x	x	-	-	-
[93]	x	-	-	-	-
[36]	x	-	-	-	-
[53, 54]	x	-	-	-	-
[37, 38]	x	-	-	-	-

Table 5. Research on SoD Enforcement in the RBAC and ABAC model and the supported SoD properties

The publications dealing with enforcement of SoD in workflows all use an underlying RBAC model. While Botha and Eloff [14] establish a mapping to RBAC in order to activate roles and directly enforce (Simple Dynamic) SoD, the works of Schaad et al. [84] and Mendt et al. [59] propose model checking approaches that cover the more advanced SoD properties. Therefore, Schaad et al. use a general model-checking approach in combination with a model checking tool (NuSMV) while Mendt et al. use a probabilistic model checker to address uncertain information during the execution of the workflow. The supported SoD properties are listed in Table 6.

[15]The full specification of the AuthUML language with all predicates and rules can be found in [3].

Pub.	Static	Simple Dynamic	Object Based	Operational Based	History Based
[14]	x	x	-	-	-
[84]	x	x	x	x	x
[59]	x	x	-	x	x
[10]	x	x	-	-	-

Table 6. Research on SoD Enforcement in workflows and the supported SoD properties

The works concerning the enforcement of SoD in a specific Context cannot be compared directly, as they deal with different enforcement contexts. In Table 7 we listed the publications, the supported SoD properties, as well as the respective context. Additionally, it is notable that all works rely on RBAC as access model.

Pub.	Enforcement context	Static	Simple Dynamic	Object Based	Operational Based	History Based
[16]	non-centralised env.	x	x	-	-	-
[91]	web-services	x	x	x	x	x
[18]	location-based	x	x	-	-	-
[23]	ad-hoc collaborations	x	-	-	-	-
[110]	multi-domain env.	x	-	-	-	-

Table 7. Research on SoD Enforcement in workflows and the supported SoD properties

The early research on SoD Enforcement in databases [7, 64, 81] can be seen as pioneering efforts that are based on each other. Their key objective was to support History Based SoD, although that term was not yet introduced at the time. While Sandhu et al. [7, 81] use TCEs, the work of Nyanchama and Osborn [64] combines object-oriented databases with roles. The later publication by Asghar et al. [8] addresses databases in cloud environments. Their key objective is to hide attributes necessary for authorization from a cloud provider. The works on SoD Enforcement in databases support all types of SoD properties (cfg. Table 9).

Pub.	Static	Simple Dynamic	Object Based	Operational Based	History Based
[7, 81]	x	x	x	x	x
[64]	x	x	x	x	x
[8]	x	x	x	x	x

Table 8. Research on SoD Enforcement in workflows and the supported SoD properties

Pub.	Static	Simple Dynamic	Object Based	Operational Based	History Based
[89]	x	x	x	x	x
[83]	x	x	x	x	-
[29]	x	-	-	-	-
[3]	x	x	x	x	x
[62]	-	-	x	-	-

Table 9. Research on SoD Enforcement in workflows and the supported SoD properties

The works of Schaad [83] and Simon and Zurko [89] both provide implementations for SoD conflict detection in RBAC. In [83] Prolog is used as programming language and the focus lies on conflict detection when delegating roles.

The authors also provides the source code of the program. Simon and Zurko [89], on the other hand, describe the implementation within the adage toolkit. They introduce the SoD properties with their implementation specifically focusing on achieving these properties. However, neither paper addresses runtime performance, which we believe is a crucial consideration given the substantial amount of data involved in IAM.

The other studies focus on different aspects in the field of SoD software developement: Nash et al. [62] address implementation considerations related to Object-Based SoD, Fong et al. [29] propose an annotation-based extension for Java, and Alghathbar [3] explore SoD in the context of requirements engineering.

7 SOD ANALYSIS

In the following, we present literature that proposes approaches to analyse or transform SoD policies. Current literature considers SoD in the context of MERs, role mining, and workflows. Additionally, transformation algorithms (e.g. transforming a set of default permission-based SoD constraints to MERs) are presented.

7.1 Problems and algorithms in context of MERs

Schefer et al. [87] and Strembeck and Mendling [94] propose two related studies that analyse and discuss conflicts of process-related mutual-exclusion (i.e. SoD) and BoD constraints. BoD denotes that a user who performed a particular task is also responsible for executing another task e.g. "a doctor who has examined the patient must also write the diagnosis". An obvious conflict for SoD and BoD constraints would be that SoD specifies two tasks as mutual exclusive while BoD states that the task must be performed by the same user. This is referred as "Direct Static Mutual Exclusion conflict" in [87, 94]. A possible solution could be, to remove the SoD constraint or to apply the BoD constraint on a role-level instead of a subject-based level. Similar to this example, the authors present the following conflicts in their work:

- **Self-contraint conflicts:** SoD and BoD rule on the same task (see example above).
- **Ownership conflicts:** Creating a SoD constraint on two tasks that are already assigned to a role.
- **Transitive constraint conflicts:** Two consecutive BoD rules (task 1 and task 2) and (task 2 and task 3) and one SoD rule (task 1 and task 3) create a conflict. The BoD rules enforce the user who executes task 1 also to execute task 2 and task 3 but executing task 1 and task 3 by the same user is prohibited by the SoD rule.
- **Assignment conflicts:** When the assignment of roles results in SoD conflicts. This includes the assignment of roles to users as well as the creation of junior and senior roles.

Additionally, they provide several possible solutions for each conflict as well as algorithms for conflict-detection and resolution. Their algorithms are based on a so-called process-related RBAC model, which represents an RBAC model in a workflow context. Roles are assigned to tasks and MER policies are specified on a task-level rather than a role-level (a more formal definition can be found in the paper). The objectives of the eight algorithms (A1-A8) are briefly summarized below:

- Check if a static (A1) or dynamic (A2) MER constraint can be created between two tasks.
- Check if a subject binding (A3) or role binding (A4) constraint can be created between two tasks.
- Check if a role can be assigned to a task (A5), another senior role (A6) or to a user (A7).
- Check if a task instance (i.e. a particular task in a running workflow) can be assigned to a user (A8).

The so-called SMER-MIN-USER problem [76] is defined as the problem to detect the minimum number of users required to ensure that each role is assigned to at least one user and that no MER constraint is violated. The purpose of

addressing the problem is to ensure operational continuity by having every role in the organization covered by at least one user. Roy et al. [76] state that a simple brute-force approach is computationally complex and present a generic search algorithm for solving that problem.

A related challenge for analysing SoD in the context of mutually exclusive roles and workflows is the Employee Assignment Problem studied in [77]. The problem tries to maximize the amount of user-to-role assignments[16] while staying compliant with SoD constraints, expressed as t-m SMER constraints. This problem seeks to maximize flexibility in the organization by enabling as many users as possible to perform different roles.

In [77] the problem is proven to be NP-complete and an optimal solution using linear programming as well as a heuristic approach is provided. Building upon the need for maximizing flexibility in the organization the employee-replacement problem is defined and studied by Roy et al. [78]. The problem describes the situation that a set of users needs to be replaced by another one while complying to SoD and BoD constraints in an ABAC context. Roy et al. [78] demonstrate the problem to be NP-hard and propose and evaluate a solution that utilizes approaches from the Conjunctive Normal Form Satisfiability problem.

7.2 Role mining

A challenge that has received less attention in literature is role mining in compliance with SoD constraints. In short, RM deals with the issue of generating roles based on existing user-permission assignments (UPA).

In a first approach towards SoD compliance [55, 102], the so-called role mining problem is extended. The original role mining problem is defined as the problem of generating a set of as few roles as possible that exactly covers the UPAs. An extended version defined as the "constraint-aware role mining problem" is studied in [102] and [55].

The constraint-aware role mining problem allows the generation of negative user-role and role-permission assignments. These negative assignments explicitly remove permissions from the user even if the user would otherwise get these permission via another role. This allows the generation of simple SoD requirements and constraints, e.g. "John should not have access to permissions of the marketing role". Note that this approach is not suitable for taking already existing constraints, e.g. mutual exclusive permissions into account, but rather generates constraints in order to further minimize the number of roles.

The role mining problem in the presence of already existing SoD requirements is described in [82]. It considers the basic role mining problem and allows an additional set of default permission-based SoD constraints as input. The goal is to generate as few roles as possible that cover all existing UPAs and additionally generate a set of t-m SMER constraints that enforce the inserted permission-based constraints. A first approach is to re-use already existing unconstrained role mining algorithms and post-process the result, i.e. generate the role constraints afterwards. For the post-processing, three algorithms are proposed: the approach of Li et al. [50] (see section 7.4 of this work), an algorithm that only creates mutual exclusive roles and an algorithm that generates constrained role-sets in which no user is allowed to be a member of all roles (t-t SMER constraints). The approach of Li et. al [50] generates the optimal constraint set but is computational expensive while the other two algorithms may yield in constraints that are more restrictive than necessary. This results in the proposal of a fourth algorithm that considers the SoD constraints already during the initial role mining phase. Basically, this SoD-aware role mining algorithm generates the roles in a way, that the algorithm of Li et al. [50] can be executed more efficiently. In a final evaluation, Sarana et al. [82] show that this approach yields in good results while still performing good.

[16] It might be noted, that this may violate the Principle of Least Privilege [27, 80], which states that a user should have only the roles that are required to do her job.

7.3 Workflow Analysis

Several works address the analysis of workflows in the context of SoD. A comprehensive approach, consisting of a specification language for expressing SoD in workflow environments, formal specifications for workflow constraint consistency, and algorithms for consistency checking is proposed by Bertino and Ferrari [11]. Their work focuses on the assignment of users and roles to tasks in a workflow and considering Static and Dynamic SoD. Their overall approach for workflow execution and constraint checking consists of different phases. At first, a workflow role specification (i.e., which roles are allowed to execute which tasks) and the constraints are designed and checked in the static analysis phase. This is a basic consistency check for the workflow design and if it fails, the workflow specification or the constraints must be reworked. An example for an inconsistency at that phase would be, if one constraint states that role r must execute task t (Binding of Duty), while another constraint states that r is not allowed to execute t. After the static analysis a pruning and consecutive planning phase are executed. The pruning reduces the complexity of the workflow (e.g. by eliminating redundant constraints) and the planning phase finally generates a set of role and user assignments to the task of the workflow. These assignments are then used in the run-time phase in order to ensure a conflict-free workflow execution. If a task cannot be executed during the run-time phase by a specific user or role for any reason (e.g., illness of the user), the planning-phase can be repeated. Bertino and Ferrari [11] provide algorithms for the static analysis, the pruning, and planning phase as well as the run-time phase.

In a similar work, Liu et al. [52] analyse so-called duty-conflict relationships between tasks to enforce Static and Dynamic SoD in role-based workflow environments. In their model, a task contains a set of task-related permissions and when a task is assigned to a user or to a role, the duty to execute that task as well as the contained permissions are assigned to the respective user or role. They focus on two types of duty-conflicts called duty-balancing tasks and duty supervision tasks.

A duty-balancing task represents a review task. That means, one user reviews the work of another user while both users are at the same hierarchical position and could respectively be replaced by each other. A supervision task is a task, in which a supervisor oversees the execution of another task. Therefore the users are at different hierarchical positions and cannot be replaced by each other. In both cases there is a duty-conflict, i.e., the tasks must not be carried out by the same user. The authors propose an authorization model, consisting of a planning phase and a run-time phase, which handles the assignment of tasks to users and roles, and therefore enforces the duty-conflicts.

For the planning phase two algorithms are proposed: the first takes a workflow and a set of valid roles as input and calculates a set of valid role-task planes (i.e. role-task assignments) and the second algorithm takes a role-task plan as input and calculates a user-role-task plan that describes which user should perform which tasks. As more than one possible plan might exist, this information can be stored in advance for the run-time phase in which an algorithm can change the plan (e.g. when a user gets unavailable). The feasibility of the approach is also shown by implementation of an authorization application.

Another similar algorithmic approach is proposed by Hewett and Kijsanayothin [33]. Given a workflow and a set of mutually exclusive roles in an RBAC environment, their algorithm returns a sequence of possible user-role-activations with respect to SoD. Beside enforcing SoD by complying to the calculated user-role-activation plan, their algorithm is also capable of calculating if the workflow can be satisfied (i.e., executed) by the given RBAC state.

Another way for specifying and analysing workflows and SoD is the approvability graph, introduced by Solworth [92]. Nodes in the graph represent states in the workflow and the directed edges represent actions. For the edges, "different user" constraint pairs can be specified, meaning that the pair of edges must be executed by different users.

The edges can also be assigned to roles. Graphically, constraints are represented by a red arch between the edges. Based on this graph, properties are defined to analyse the constraint workflow. An approvability graph is called *u-approvable* if, at any time during workflow execution, every permitted user (i.e. the user is in the respective role and does not have a SoD conflict) can execute the next action without the workflow getting stuck in future. Note that a workflow might get stuck, if there is no user without a SoD conflict in the role defined regarding the current action. An approvability graph is called *s-approvable* if there are paths of users performing actions that lead to the final state of the graph, but not every path allows the workflow to be executed until the end. Finally, an approvability graph is called *well-formed* if it can always complete with a sufficient number of users per role. This becomes particularly problematic when the graph has cyclic parts in it. Properties of those cyclic graphs are further analysed by Solworth [92].

A similar challenge is addressed in the work of Thipse and Hewett [96] in an RBAC context. Given a workflow, user-role assignments and a set of MERs, the presented algorithm checks if every path of the workflow can complete without violating a MER policy.

7.4 Constraint Generation and Transformation

Constraint generation and transformation in the context of static mutually exclusive roles (SMER) has been studied by Li et al. [17, 50]. They emphasise the difference between SoD policies (i.e., default permission-based SoD constraints), as overall objectives, and t-m SMER constraints, as mechanisms to enforce these objectives in RBAC systems.

Li et al. [50] show that the direct enforcement of the SoD policies in an RBAC system is computationally complex and slow (coNP-complete) while the enforcement of SMER constraints is efficient. Therefore, one goal is to compute SMER constraints that enforce SoD policies and are as less additionally restrictive as possible. To translate the SoD policies into SMER constraints, Li et al. [50] first calculate so-called role-based SoD policies, which are role-based pendants to the permission-based SoD policies. In a second step, they propose a function called SMER-Gen, which generates all sets of SMER constraints that are minimal (i.e., least restrictively) enforcing a set role-based SoD policies. It might be noted that the work of Li et al. [50] explores several properties and formal proofs of this topic that extend beyond the scope of this paper.

Addressing the same issue, Chen et al. [17] propose two constraint generation algorithms for RBAC in order to transform SoD policies into sets of SMERs. Given a set of SoD policies and an RBAC state (including role hierarchies) the algorithms are capable of generating SMERs that satisfy these SoD policies. The aim of the approach is to generate proposals, that are as little restrictive as possible. Therefore, they propose two algorithms: the first one starts with the most restrictive SMER sets and then tries to weaken or remove constraints, while staying compliant with the SoD policies. The second one gradually strengthens a constraint set until it finally complies. Both algorithms generate several proposals as the final choice usually includes criteria like role semantics and expressiveness.

Kijsanayothin et al. [42] propose a heuristic approach in order to make suggestions for possible MER policies based on a given workflow and RBAC model. In their approach, every node of a workflow consists of a triple (role, action, object). The key idea of their MER generation is, that in a workflow, an object should neither be created and later updated by the same user nor should it be updated twice by the same user. Therefore, their algorithm scans all possible workflow paths for nodes with consecutive create and update or update and update actions on the same object. The roles of these nodes are then suggested to be mutual exclusive.

Alm and Illig [4] present an approach for an implementation to translate the high-level ORKA policy language (OPL) [5] to XACML. Their goal is to combine the advantages of both languages. OPL, based on the RBAC model, is designed to be easy to use on a high level. Its administration tool comes with features for policy design (e.g. consistency checks).

XACML, on the other hand, is flexible and widely supported by tools in the industry. Their approach is capable of translating various types of SoD as well as other RBAC policies like Chines wall or BoD.

7.5 Discussion of Research on SoD Analysis

The related studies of Schefer et al. [87] and Strembeck and Mendling [94] propose several algorithms for BoD and SoD constraints in the context of tasks and MERs. Their work is valuable during the modelling process of tasks, SoD or BoD constraints as well as for conflict detection at runtime. Note that, as the work deals with MERs in workflows, it could potentially also be assigned to the workflow category in our classification scheme. The three works by Roy et al. [76–78] show some similarities. Each study defines a specific problem (i.e. MER-MIN-USER problem [76], Employee Assignment Problem [77], and the employee-replacement problem [78]) and proposes an algorithmic solution. However, the practical relevance of these problems needs to be discussed. While the SMER-MIN-USER problem [76] appears applicable as it ensures operational continuity, the Employee Assignment Problem [77] conflicts with the Principle of Least Privilege [27, 80]. Additionally, the Employee Replacement Problem [78] raises ethical concerns, as it focuses on reducing the number of employees.

Role mining in context of SoD is a topic that has received little attention up to now. The two related studies by Uzun et al. [102] and Lu et al. [55] propose an approach that generates negative assignments. However, this approach is not designed to generate roles without SoD violations but to produce constraint roles (i.e. roles with inherent constraints/ negative assignments). The only comprehensive work in this area is the work of Sarana et al. [82], in which algorithms for role mining under SoD constraints are proposed and evaluated (including the run-time performance). Nevertheless, further research is needed in this field. Future work could expand already existing approaches to SoD and consider common role mining metrics like the weighted structural complexity [60].

Regarding workflows, there are several studies that focus on planning task execution and role activation prior to run-time to ensure that tasks and roles are sequenced in a way that no SoD (or similar) constraints are violated [11, 33, 52]. The comprehensive work by Bertino et al. [11] represents the first foundational study in this area. It addresses both SoD and BoD conflicts, and the plan created can be adjusted at runtime in response to changes. Lui et al. [52] extend the range of considered constraints to include additional variations of duty conflicts (i.e. duty-balancing tasks and duty-supervision tasks). Hewett and Kijsanayothin's [33] approach is very similar to that of Bertino et al.. However, they utilize a set-based approach (instead of a logic-based one) and address various SoD properties(e.g. Object-Based, History-Based, etc.).

Finally, generating and converting SoD constraints is an important topic in this specific part of SoD research. Li et al. [50] and Chen et al. [17] demonstrate that enforcing permission-based SoD constraints can be computational complex and expensive. However, converting permission-based SoD constraints into SMER constraints results into stricter rules. Therefore, this problem cannot be solved solely through an algorithmic approach but instead requires organizational solutions. Future research might include studies that explore the impact of such transformations on real-world companies.

8 HIGH-LEVEL SOD AND MANAGEMENT

In the following, literature that examines SoD from a high-level or management perspective is presented. Engdahl [25] particularly focuses on the principle of duality in SoD, also known as the "two-man rule" or "four-eyes principle," which involves having one user's activity reviewed by another. Engdahl argues that the practical operation and effectiveness of the duality principle have not been studied empirically or theoretically, and that it is largely based on common sense.

However, the principle relies on the idea that users are willing to question the work of their colleagues and to report instances of misconduct and crime when discovered. This heavily depends on social factors. Users could neglect their auditing duties in order to avoid conflicts with colleagues or to secure their career prospects. Additionally, isolated groups or teams tend to protect each other when irregularities occur. In order to ensure that the principle of duality works, the author requires two conditions to be met:

- **Distance - relative autonomy through third-party dependence**: The two users should be independent from each other. However, they should be dependent from a third party, from which they derive their support and authority while executing their responsibility of monitoring each other.
- **Closeness - the principle of duality and interdependent work task**: The principle of duality should not be implemented by assigning one user to perform a task and another one to review the activity. Instead, both users should be required to carry out the task.

The conditions are argumentatively derived from existing literature and criminal cases from the Swedish banking and finance sector, which also serve to underline their necessity.

Wolf and Gehrke [107] consider SoD as a major class of control activities and an essential component of an organizations' internal controls, particularly for the reliability of financial reporting. They propose a method consisting of six activities in order to identify SoD conflicts. Moreover, they show how SoD rules can be derived. For each activity they describe the outcomes and potential techniques that can be used (e.g. interviews, questionnaires, SoD matrix, etc.). In contrast to approaches that simply enforce SoD rules, their method is a top-down approach that starts with high-level processes and derives SoD rules from them. The six activities are listed and briefly described below:

- **Process Scoping**: Identify high-risk processes, which will be the basis for further SoD analysis.
- **Process Analysis**: Gain a comprehensive understanding of the identified processes.
- **SoD Ruleset Specification**: Define process activities that pose a risk of fraud if carried out by a single individual (e.g. create and maintain vendor). The potential risk is documented as well.
- **SoD Ruleset Translation**: Translate the process oriented "high-level" policies to technical, system-dependent rules consisting of specific system permissions, transaction codes, function codes, authorization objects, etc..
- **Live Data Extraction and Transformation**: Extract and standardize the current users' access rights from the relevant systems. Transformation and standardization might be necessary because some applications might store the access rights in a complex manner.
- **SoD Analysis**: Determine existing SoD conflicts based on the SoD rule set and the extracted user access rights.

Wolf and Gehrke's work also demonstrates the application of the six activities in a real-world project.

Omland et al. [68] introduce an initial Maturity Model for SoD. The purpose of a Maturity Model is to provide a framework to rate the maturity level for organizations and outline the necessary steps to progress to the next level of maturity. To assess the SoD maturity of an organization, 31 questions are developed and classified into the four categories "rule set", "control processing", "SoD reporting", and "organizational framework". Based on the answers, an organization is assigned to one of the five maturity levels described below:

- **Level 0 - None**: Organization does not consider SoD.
- **Level 1 - Initial**: Unstructured processes, non-formalized, heterogeneous rule sets. Rules are defined ad-hoc and do not consider risk assessments.
- **Level 2 - Repeatable**: Formalized rule sets, transparency and involvement of departments. Conflicts are resolved and the results are documented and reported.

- **Level 3 - Defined**: Rules for all high-risk business processes and regular reassessment to rule sets. Detective and preventive controls as well as a risk-based prioritization of follow-up activities. A comprehensive and structured SoD approach.
- **Level 4 - Managed**: Generic and organization-wide uniform rule set, and risk-based definition of SoD for all relevant processes. Specific rules for applications are derived from the generic rule set. Automated escalation management and inclusion of process improvement metrics in reports.
- **Level 5 - Optimizing**: Complete, organization-wide uniform rule set and well-defined processes for updating the rule set. Continuous development and improvement of SoD processes. Innovative processes and technology improvements are enabled.

In the context of financial reporting, Elsas [24] in cooperation with Deloitte Touche Tohmatsu International focuses on the auditors' view of SoD. In order to detect potential fraud scenarios, they use a so-called "Supercycle diagram" which depicts the cash and resource flows within an organization. The diagram is then transformed into two matrices: a flow matrix that captures the cash and resource flows, and an authorization and ability matrix that provides details about authorized users and the nature of their authorizations, such as withdrawal, addition, or interaction. These matrices serve as the foundation for generating potential solo-fraud scenarios and proposing solutions to address the conflicts. It is emphasized that the scenarios might be subtle but sometimes hard to detect, such as a financial administrator with the ability to withdraw credit. As a result, these scenarios significantly reduce the workload for auditors.

A model for separating duties within an organization combining theoretical, pedagogical, and practical literature is developed by Kobelsky [104]. The model distinguishes between primary SoD, which focuses on error detection and fraud prevention and secondary SoD, which focuses on maintaining a consistent and repeatable level of internal control. The primary model defines the duties "Custody, Valuation, Decision-Making", "Recording", "Primary Authorization", and "Reconciliation". It suggests segregating an activity between two users. While one user can perform and record a "Custody, Valuation, Decision-Making" activity, another user needs to give the authorization for that activity and reconcile the report and the activity. Consequently, the secondary SoD has to ensure that the auditing duties from primary SoD (authorization and reconciliation) are performed correctly[17].

Pankowska [71] proposes a business multi-model for SoD management. The multi-model describes a business architecture with a focus on SoD and consists of five elements: business goals, functions, and active elements (1), critical processes (2), contracts (3), SoD (4), and SoD application for access rights distribution and assurance mapping (5). To define roles and responsibilities within SoD the CERI model is proposed consisting of the roles Controller (C), Executor (E), Receptionist (R), and Investigator (I). In the CERI model the Controller establishes SoD regulations, the Executor carries out tasks or processes, the Receptionist is responsible for information processing, and the Investigator is responsible for SoD validation and verification.

8.1 Discussion of Research on SoD Management

The works of Engdahl [25], Wolf and Gehrke [107], and Elsas [24] demonstrate that SoD is an organizational challenge that cannot be solved solely via technical or algorithmic approaches. However, up to now there are very few approaches that address the management of SoD at a higher, possibly organizational level. For example, there are several standards that organizations must adhere to [107], and it must be ensured that the SoD policies within a company align with these standards. Therefore, in addition to exploring more management approaches, it has become essential that algorithmic

[17] Note that this is only a brief description of the model.

methods, such as the SMER transformation by Li et al. [17, 50], are evaluated against real-world requirements. This may result in hybrid approaches that combine algorithms with management and process models.

9 DISCUSSION

After answering RQ1 ("What is the relevant literature in SoD?") and RQ2 ("How can the stream of relevant literature in SoD be structured?") we now answer RQ3 ("What is the state of the art and what are the current challenges for SoD?").

When analysing the publication dates (Figure 2), one can see, that the early period of SoD research until 1993 is dominated by papers focusing on the enforcement of SoD rules. This is the first phase of early research driven by the access control community that focuses on the enforcement of SoD in databases. About a decade later, with the increasing popularity of the access models (especially the RBAC model) more research in the area of SoD Specification was carried out. Consequently, a few years later (about 2005/ 2006), a rapid diversification of research towards papers focusing on enforcement and analysis can be observed. This development seems reasonable, as enforcement and analysis can only be done after specification. The popularity of access model research during this time also explains the peak of total published papers in the years 2006-2008. Afterwards, research activities decrease, also corresponding to access control research. When analysing the papers of the following years 2009 until 2023, three major trends in the streams of literature, that are regularly addressed, can be detected.

- **Access control models** [4, 8, 9, 12, 22, 26, 32, 37, 38, 41, 53–55, 63, 76–78, 82, 100, 102, 103, 110, 112]: Research in the area of access control models is continued and modern challenges are addressed. The ABAC model, sometimes refereed to the successor of the RBAC model, is increasingly being studied. In addition, there are several recent works that use Petri nets to simulate access control and enforce SoD. The research in the field of access control still constitutes the largest part of SoD research.
- **Workflows** [10, 59, 87, 94, 111]: Workflows are regularly addressed in SoD research. A common challenge is the creation of workflow execution plans and replanning while considering SoD rules. Note, that there is some research in the workflow community that focuses on the WSP, that we have explicitly excluded from this study (see Section 4.2).
- **High-level SoD & management** [25, 68, 71, 104, 107]: In general, it can be observed that research output dealing with high-level SoD and management has grown since 2009. This can be related to the increasing number of organizations that implement SoD as well as the increasing number of legal requirements for organizations (e.g. the Sarbanes-Oxley Act [65], the BSI Grundschutz [66] or the American Health Insurance Portability and Accountability Act [101]).

Finally, based on the analyses upon, we discuss some current challenges and potential research gaps. Regarding access models, most research is dedicated to the RBAC model. However, the ABAC model is a more advanced model and the successor of RBAC. Only few research has addressed the model so far (e.g. [9, 37, 38]) and there are still open issues like the enforcement under different conditions and circumstances, the combination of ABAC and workflows, or maintaining ABAC policies in context of SoD. Especially, data quality plays a major role in the context of ABAC [47].

When maintaining SoD rules, human understandability is an important topic. It can be achieved by adding visualizations or semantics to SoD rules. Visualization of SoD policies is currently only addressed in research as a byproduct of the SoD modelling language research or within the Adage toolkit [89]. Additionally, current research assumes that the SoD rules are already pre-defined and existing. So far, only one paper proposes a process for creating (meaningful) SoD rules [107]. Finally, more research regarding social aspects (like the work presented by Engdahl [25]) is necessary. SoD

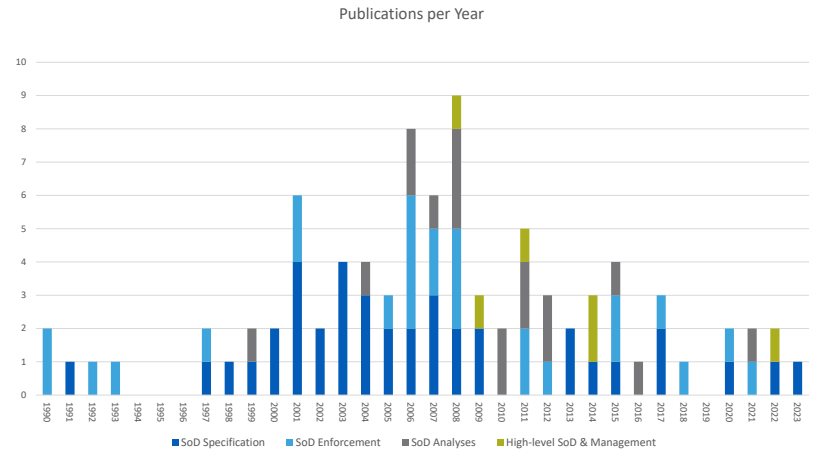


Fig. 2. Relevant SoD Publications per Year

needs to be comprehensively integrated into an organization i.e. in the management, in the business processes, on a technical level, and, not at least, into the minds of the users. Recent research mainly explores the enforcement on a technical level. However, the effectiveness of SoD as a security measure still relies on human factors and mutual trust.

10 CONCLUSION

In this work, we conducted a comprehensive literature review in the area of SoD research. Proceeding according to the Eight-Step Guide of Okoli and Schabram [67], we defined the three research objectives RO1 ("Identify the relevant literature in the field of SoD"), RO2 ("Structure the body of relevant literature in the field of SoD"), and RO3 ("Elaborate on the state of the art and identify the current challenges in the field of SoD").

RO1 is addressed by conducting a structured database search and filtering the relevant results according to our defined filter steps. Subsequently, we tackle RO2 by developing a multi-level classification scheme and categorizing the relevant literature. After introducing the classification scheme in general, every one of the 19 categories together with the assigned 88 publications is described. Finally, we address RO3 by proposing three major trends in the streams of literature and discussing current challenges and potential research gaps. We show that the technical specification and implementation of SoD rules have already been addressed extensively. However, current research lacks, amongst others, visualisations, the semantic of SoD rules, and the exploration of social factors.

REFERENCES

[1] Gail-Joon Ahn and Ravi Sandhu. 1999. The RSL99 Language for Role-Based Separation of Duty Constraints. In *Proceedings of the Fourth ACM Workshop on Role-Based Access Control* (Fairfax, Virginia, USA) *(RBAC '99)*. Association for Computing Machinery, New York, NY, USA, 43–54.

[2] Gail-Joon Ahn and Ravi Sandhu. 2000. Role-Based Authorization Constraints Specification. 3, 4 (nov 2000), 207–226. https://doi.org/10.1145/382912.382913

[3] Khaled Alghathbar. 2007. Validating the enforcement of access control policies and separation of duty principle in requirement engineering. *Information and Software Technology* 49, 2 (2007), 142–157. https://doi.org/10.1016/j.infsof.2006.03.009

[4] Christopher Alm and Roland Illig. 2010. Translating High-Level Authorization Constraints to XACML. In *2010 6th World Congress on Services*. 629–636. https://doi.org/10.1109/SERVICES.2010.62

[5] Christopher Alm, Ruben Wolf, and Joachim Posegga. 2009. The OPL Access Control Policy Language. In *Trust, Privacy and Security in Digital Business*, Simone Fischer-Hübner, Costas Lambrinoudakis, and Günther Pernul (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 138–148.

[6] American National Standards Institute. 2004. Role Based Access Control. ANSI/INCITS 359-2004.

[7] P. Ammann and R. Sandhu. 1992. Implementing transaction control expressions by checking for absence of access rights. In *Proceedings Eighth Annual Computer Security Application Conference*. IEEE Computer Society, Los Alamitos, CA, USA, 131,132,133,134,135,136,137,138,139,140. https://doi.org/10.1109/CSAC.1992.228226

[8] Muhammad Rizwan Asghar, Giovanni Russello, and Bruno Crispo. 2015. E-GRANT: Enforcing Encrypted Dynamic Security Constraints in the Cloud. In *2015 3rd International Conference on Future Internet of Things and Cloud*. 135–144. https://doi.org/10.1109/FiCloud.2015.43

[9] Nuray Baltaci and James Joshi. 2020. A Constraint and Risk-aware Approach to Attribute-based Access Control for Cyber-Physical Systems. *Computers Security* 96 (05 2020), 101802. https://doi.org/10.1016/j.cose.2020.101802

[10] David Basin, Samuel J. Burri, and Günter Karjoth. 2012. Dynamic Enforcement of Abstract Separation of Duty Constraints. 15, 3, Article 13 (nov 2012), 30 pages. https://doi.org/10.1145/2382448.2382451

[11] Elisa Bertino, Elena Ferrari, and Vijay Atluri. 1999. The Specification and Enforcement of Authorization Constraints in Workflow Management Systems. 2, 1 (feb 1999), 65–104.

[12] Khalid Bijon, R. Krishman, and R. Sandhu. 2013. Constraints specication in attribute based access control. *SCIENCE* 2 (01 2013), 131–144.

[13] Ferdinand Bollwein and Lena Wiese. 2017. Closeness Constraints for Separation of Duties in Cloud Databases as an Optimization Problem. In *Data Analytics*, Andrea Calì, Peter Wood, Nigel Martin, and Alexandra Poulovassilis (Eds.). Springer International Publishing, Cham, 133–145.

[14] R. A. Botha and J. H. P. Eloff. 2001. Separation of duties for access control enforcement in workflow environments. *IBM Systems Journal* 40, 3 (2001), 666–682. https://doi.org/10.1147/sj.403.0666

[15] Sara Brockmans, Raphael Volz, Andreas Eberhart, and Peter Löffler. 2004. Visual Modeling of OWL DL Ontologies Using UML. In *The Semantic Web – ISWC 2004*, Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 198–213.

[16] David W Chadwick, Wensheng Xu, Sassa Otenko, Romain Laborde, and Bassem Nasser. 2007. Multi-session Separation of Duties (MSoD) for RBAC. In *2007 IEEE 23rd International Conference on Data Engineering Workshop*. 744–753. https://doi.org/10.1109/ICDEW.2007.4401062

[17] Hong Chen and Ninghui Li. 2006. Constraint Generation for Separation of Duty *(SACMAT '06)*. Association for Computing Machinery, New York, NY, USA, 130–138. https://doi.org/10.1145/1133058.1133077

[18] Weihe Chen, Zhu Tang, and Shiguang Ju. 2008. Enforcement of Spatial Separation of Duty Constraint. In *2008 The 9th International Conference for Young Computer Scientists*. 2108–2114. https://doi.org/10.1109/ICYCS.2008.223

[19] David D. Clark and David R. Wilson. 1987. A Comparison of Commercial and Military Computer Security Policies. In *1987 IEEE Symposium on Security and Privacy*. 184–184. https://doi.org/10.1109/SP.1987.10001

[20] Jason Crampton. 2003. Specifying and Enforcing Constraints in Role-Based Access Control. In *Proceedings of the Eighth ACM Symposium on Access Control Models and Technologies* (Como, Italy) *(SACMAT '03)*. Association for Computing Machinery, New York, NY, USA, 43–50.

[21] Jason Crampton. 2005. A Reference Monitor for Workflow Systems with Constrained Task Execution *(SACMAT '05)*. Association for Computing Machinery, New York, NY, USA, 38–47. https://doi.org/10.1145/1063979.1063986

[22] Marcelo Antonio de Carvalho and Paulo Bandiera-Paiva. 2017. Evaluating ISO 14441 privacy requirements on role based access control (RBAC) restrict mode via Colored Petri Nets (CPN) modeling. In *2017 International Carnahan Conference on Security Technology (ICCST)*. 1–8. https://doi.org/10.1109/CCST.2017.8167833

[23] Lingli Deng, Yeping He, and Ziyao Xu. 2008. Enforcing Separation of Duty in Ad Hoc Collaboration. In *2008 The 9th International Conference for Young Computer Scientists*. 1545–1552. https://doi.org/10.1109/ICYCS.2008.131

[24] Ph.I. Elsas. 2008. X-raying Segregation of Duties: Support to illuminate an enterprise's immunity to solo-fraud. *International Journal of Accounting Information Systems* 9, 2 (2008), 82–93. https://doi.org/10.1016/j.accinf.2007.10.005 2007 Research Symposium on Information Systems Assurance.

[25] Oskar Engdahl. 2014. Ensuring regulatory compliance in banking and finance through effective controls: The principle of duality in the segregation of duties. *Regulation & Governance* 8, 3 (2014), 332–349. https://doi.org/10.1111/rego.12027 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/rego.12027

[26] M. Esna-Ashari, H.R. Rabiee, and S.H. Mirian-Hosseinabadi. 2011. Reliability of separation of duty in ANSI standard role-based access control. *Scientia Iranica* 18, 6 (2011), 1416–1424. https://doi.org/10.1016/j.scient.2011.08.016

[27] David Ferraiolo and David Kuhn. 1992. Role-Based Access Controls. 15th National Computer Security Conference, Baltimore, MD.

[28] S. Foley. 1991. Separation of duty using high water marks. In *Proceedings Computer Security Foundations Workshop IV*. IEEE Computer Society, Los Alamitos, CA, USA, 79,80,81,82,83,84,85,86,87,88. https://doi.org/10.1109/CSFW.1991.151572

[29] Philip W. L. Fong. 2006. Discretionary Capability Confinement. In *Computer Security – ESORICS 2006*, Dieter Gollmann, Jan Meier, and Andrei Sabelfeld (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 127–144.

[30] Virgil D Gligor, Serban I Gavrila, and David Ferraiolo. 1998. On the formal definition of separation-of-duty policies and their composition. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186)*. IEEE, 172–183.

[31] Gregory Gutin, Stefan Kratsch, and Magnus Wahlström. 2016. Polynomial kernels and user reductions for the workflow satisfiability problem. *Algorithmica* 75, 2 (2016), 383–402.

[32] Muhammad Asif Habib, Nasir Mahmood, Muhammad Shahid, Muhammad Umar Aftab, Uzair Ahmad, and C. Muhammad Nadeem Faisal. 2014. Permission based implementation of Dynamic Separation of Duty (DSD) in Role based Access Control (RBAC). In *2014 8th International Conference*

on Signal Processing and Communication Systems (ICSPCS). 1–10. https://doi.org/10.1109/ICSPCS.2014.7021054

[33] Rattikorn Hewett and Phongphun Kijsanayothin. 2008. Authorization constraint enforcement for information system security. In *2008 IEEE International Conference on Systems, Man and Cybernetics*. 3502–3507. https://doi.org/10.1109/ICSMC.2008.4811840

[34] Charles Antony Richard Hoare et al. 1985. *Communicating sequential processes*. Vol. 178. Prentice-hall Englewood Cliffs.

[35] Vincent Hu, David Ferraiolo, D. Kuhn, A. Schnitzer, Knox Sandlin, R. Miller, and Karen Scarfone. 2014. Guide to attribute based access control (ABAC) definition and considerations. *National Institute of Standards and Technology Special Publication* (01 2014), 162–800.

[36] Jun-Cheol Jeon and Kee-Young Yoo. 2006. Conflict Detection in Role-Based Access Control Using Multiple-Attractor Cellular Automata. In *Intelligent Computing*, De-Shuang Huang, Kang Li, and George William Irwin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 533–541.

[37] Sadhana Jha, Shamik Sural, Vijayalakshmi Atluri, and Jaideep Vaidya. 2015. Enforcing Separation of Duty in Attribute Based Access Control Systems. In *Information Systems Security*, Sushil Jajoda and Chandan Mazumdar (Eds.). Springer International Publishing, 61–78.

[38] Sadhana Jha, Shamik Sural, Vijayalakshmi Atluri, and Jaideep Vaidya. 2018. Specification and Verification of Separation of Duty Constraints in Attribute-Based Access Control. *IEEE Transactions on Information Forensics and Security* 13, 4 (2018), 897–911.

[39] James Joshi, Elisa Bertino, U. Latif, and A. Ghafoor. 2001. Generalized Temporal Role Based Access Control Model (GTRBAC) (Part I)Specification and Modeling. *IEEE Transactions on Knowledge and Data Engineering - TKDE* (01 2001).

[40] James B. D. Joshi, Basit Shafiq, Arif Ghafoor, and Elisa Bertino. 2003. Dependencies and Separation of Duty Constraints in GTRBAC *(SACMAT '03)*. Association for Computing Machinery, New York, NY, USA, 51–64. https://doi.org/10.1145/775412.775420

[41] Slim Kallel, Anis Charfi, Mira Mezini, Mohamed Jmaiel, and Karl Klose. 2009. From Formal Access Control Policies to Runtime Enforcement Aspects. In *Engineering Secure Software and Systems*, Fabio Massacci, Samuel T. Redwine, and Nicola Zannone (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 16–31.

[42] P. Kijsanayothin, R. Hewett, and A. Thipse. 2008. Security Analysis of Role-based Separation of Duty with Workflows. In *2008 3rd International Conference on Availability, Reliability and Security (ARES x27;08)*. IEEE Computer Society, Los Alamitos, CA, USA, 765–770.

[43] Konstantin Knorr and Henrik Stormer. 2001. Modeling and Analyzing Separation of Duties in Workflow Environments. In *Trusted Information*, Michel Dupuy and Pierre Paradinas (Eds.). Springer US, Boston, MA, 199–212.

[44] Konstantin Knorr and Harald Weidner. 2001. Analyzing Separation of Duties in Petri Net Workflows. In *Information Assurance in Computer Networks*, Vladimir I. Gorodetski, Victor A. Skormin, and Leonard J. Popyack (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 102–114.

[45] Manuel Koch and Francesco Parisi-Presicce. 2003. Visual Specifications of Policies and Their Verification. In *Fundamental Approaches to Software Engineering*, Mauro Pezzè (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 278–293.

[46] D. Richard Kuhn. 1997. Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems. In *Proceedings of the Second ACM Workshop on Role-Based Access Control* (Fairfax, Virginia, USA) *(RBAC '97)*. Association for Computing Machinery, New York, NY, USA, 23–30. https://doi.org/10.1145/266741.266749

[47] Michael Kunz, Alexander Puchta, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. 2019. Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications* 44 (2019), 64–79. https://doi.org/10.1016/j.jisa.2018.11.004

[48] JuHum Kwon and Chang-Joo Moon. 2006. Visual modeling and formal specification of constraints of RBAC using semantic web technology. *Knowledge-Based Systems* 20, 4 (2006), 350–356. https://doi.org/10.1016/j.knosys.2006.08.002

[49] Yair Levy and Timothy J Ellis. 2006. A systems approach to conduct an effective literature review in support of information systems research. *Informing Science* 9 (2006).

[50] Ninghui Li, Mahesh V. Tripunitara, and Ziad Bizri. 2007. On Mutually Exclusive Roles and Separation-of-Duty. 10, 2 (may 2007), 5–es.

[51] Ninghui Li and Qihua Wang. 2008. Beyond Separation of Duty: An Algebra for Specifying High-Level Security Policies. *J. ACM* 55, 3, Article 12 (aug 2008), 46 pages. https://doi.org/10.1145/1379759.1379760

[52] Duen-Ren Liu, Mei-Yu Wu, and Shu-Teng Lee. 2004. Role-based authorizations for workflow systems in support of task-based separation of duty. *Journal of Systems and Software* 73, 3 (2004), 375–387. https://doi.org/10.1016/S0164-1212(03)00175-4

[53] Meng Liu, Yun Luo, Chi Yang, Shaoning Pang, Deepak Puthal, Kaijun Ren, and Xuyun Zhang. 2020. Privacy-preserving matrix product based static mutual exclusive roles constraints violation detection in interoperable role-based access control. *Future Generation Computer Systems* 109 (2020), 457–468. https://doi.org/10.1016/j.future.2018.10.017

[54] Meng Liu, Xuyun Zhang, Chi Yang, Shaoning Pang, Deepak Puthal, and Kaijun Ren. 2017. Privacy-Preserving Detection of Statically Mutually Exclusive Roles Constraints Violation in Interoperable Role-Based Access Control. In *2017 IEEE Trustcom/BigDataSE/ICESS*. 502–509.

[55] Haibing Lu, Jaideep Vaidya, Vijayalakshmi Atluri, and Yuan Hong. 2012. Constraint-Aware Role Mining via Extended Boolean Matrix Decomposition. *IEEE Transactions on Dependable and Secure Computing* 9, 5 (2012), 655–669. https://doi.org/10.1109/TDSC.2012.21

[56] Jianfeng Lu, Ruixuan Li, Zhengding Lu, Jinwei Hu, and Xiaopu Ma. 2009. Specification and Enforcement of Static Separation-of-Duty Policies in Usage Control. In *Information Security*, Pierangela Samarati, Moti Yung, Fabio Martinelli, and Claudio A. Ardagna (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 403–410.

[57] Richard W. C. Lui, Sherman S. M. Chow, Lucas C. K. Hui, and S. M. Yiu. 2005. Role Activation Management in Role Based Access Control. In *Information Security and Privacy*, Colin Boyd and Juan Manuel González Nieto (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 358–369.

[58] Jan Mendling, Karsten Ploesser, and Mark Strembeck. 2008. Specifying Separation of Duty Constraints in BPEL4People Processes. In *Business Information Systems*, Witold Abramowicz and Dieter Fensel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 273–284.

[59] Tamara Mendt, Carsten Sinz, and Olga Tveretina. 2011. Analyzing Separation of Duties Constraints with a Probabilistic Model Checker. In *Business Information Systems Workshops*, Witold Abramowicz, Leszek Maciaszek, and Krzysztof Węcel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 18–29.

[60] Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. 2010. Mining Roles with Multiple Objectives. 13, 4, Article 36 (dec 2010), 35 pages. https://doi.org/10.1145/1880022.1880030

[61] T. Mossakowski, M. Drouineaud, and K. Sohr. 2003. A temporal-logic extension of role-based access control covering dynamic separation of duties. In *10th International Symposium on Temporal Representation and Reasoning, 2003 and Fourth International Conference on Temporal Logic. Proceedings.* 83–90. https://doi.org/10.1109/TIME.2003.1214883

[62] M.J. Nash and K.R. Poland. 1990. Some conundrums concerning separation of duty. In *Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy.* 201–207. https://doi.org/10.1109/RISP.1990.63851

[63] Dang Nguyen, Jaehong Park, and Ravi Sandhu. 2013. A provenance-based access control model for dynamic separation of duties. In *2013 Eleventh Annual Conference on Privacy, Security and Trust.* 247–256. https://doi.org/10.1109/PST.2013.6596060

[64] Matunda Nyanchama and Sylvia Osborn. 1993. Role-Based Security, Object Oriented Databases and Separation of Duty. *SIGMOD Rec.* 22, 4 (dec 1993), 45–51. https://doi.org/10.1145/166635.166652

[65] One Hundred Seventh Congress of the United States of America. 2002. Sarbanes-Oxley Act of 2002. Pub. L. No. 107-204, 116 Stat. 745.

[66] Federal office for information security (BSI). n.d.. IT-Grundschutz. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html. Accessed: 2023-06-10.

[67] Chitu Okoli and Kira Schabram. 2010. A Guide to Conducting a Systematic Literature Review of Information Systems Research. *SSRN Electronic Journal* 10 (05 2010). https://doi.org/10.2139/ssrn.1954824

[68] Jan Omland, Nick Gehrke, and Niels Müller-Wickop. 2011. A Maturity Model for Segregation of Duties in Standard Business Software. In *Governance and Sustainability in Information Systems. Managing the Transfer and Diffusion of IT*, Markus Nüttgens, Andreas Gadatsch, Karlheinz Kautz, Ingrid Schirmer, and Nadine Blinn (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 288–294.

[69] Klaus Ostermann, Mira Mezini, and Christoph Bockisch. 2005. Expressive Pointcuts for Increased Modularity. In *ECOOP 2005 - Object-Oriented Programming*, Andrew P. Black (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 214–240.

[70] F. Paci, J. Crampton, and E. Bertino. 2006. Access Control and Authorization Constraints for WS-BPEL. In *2006 IEEE International Conference on Web Services.* IEEE Computer Society, Los Alamitos, CA, USA, 275–284. https://doi.org/10.1109/ICWS.2006.21

[71] Małgorzata Pańkowska. 2022. Segregation of Duties in Business Architecture Models. In *Electronic Governance and Open Society: Challenges in Eurasia*, Andrei V. Chugunov, Marijn Janssen, Igor Khodachek, Yuri Misnikov, and Dmitrii Trutnev (Eds.). Springer International Publishing, 35–49.

[72] Guy Pare, Mary Tate, David Johnstone, and Spyros Kitsiou. 2016. Contextualizing the twin concepts of systematicity and transparency in information systems literature reviews. *European Journal of Information Systems* 25 (11 2016), 493–508. https://doi.org/10.1057/s4I303-016-0020-3

[73] Jaehong Park and Ravi Sandhu. 2004. The UCONABC Usage Control Model. *ACM Trans. Inf. Syst. Secur.* 7, 1 (feb 2004), 128–174.

[74] Indrakshi Ray, Na Li, Robert France, and Dae-Kyoo Kim. 2004. Using Uml to Visualize Role-Based Access Control Constraints *(SACMAT '04).* Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/990036.990054

[75] Amira Regayeg, Ahmed Hadj Kacem, and Mohamed Jmaiel. 2005. Towards a Formal Methodology for Designing Multi-agent Applications. In *Multiagent System Technologies*, Torsten Eymann, Franziska Klügl, Winfried Lamersdorf, Matthias Klusch, and Michael N. Huhns (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 153–164.

[76] Arindam Roy, Shamik Sural, and Arun Kumar Majumdar. 2012. Minimum user requirement in Role Based Access Control with Separation of Duty constraints. In *2012 12th International Conference on Intelligent Systems Design and Applications (ISDA).* 386–391.

[77] Arindam Roy, Shamik Sural, Arun Kumar Majumdar, Jaideep Vaidya, and Vijayalakshmi Atluri. 2016. On Optimal Employee Assignment in Constrained Role-Based Access Control Systems. *ACM Trans. Manage. Inf. Syst.* 7, 4, Article 10 (dec 2016), 24 pages. https://doi.org/10.1145/2996470

[78] Arindam Roy, Shamik Sural, Arun Kumar Majumdar, Jaideep Vaidya, and Vijayalakshmi Atluri. 2021. Enabling Workforce Optimization in Constrained Attribute-Based Access Control Systems. *IEEE Transactions on Emerging Topics in Computing* 9, 4 (2021), 1901–1913.

[79] R. Sandhu. 1988. Transaction control expressions for separation of duties. In *[Proceedings 1988] Fourth Aerospace Computer Security Applications.* 282–286. https://doi.org/10.1109/ACSAC.1988.113349

[80] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. 1996. Role-based access control models. *Computer* 29, 2 (1996), 38–47.

[81] Ravi S Sandhu. 1990. Separation of Duties in Computerized Information Systems.. In *DBSec.* Citeseer, 179–190.

[82] Prasuna Sarana, Arindam Roy, Shamik Sural, Jaideep Vaidya, and Vijayalakshmi Atluri. 2015. Role Mining in the Presence of Separation of Duty Constraints. In *Information Systems Security*, Sushil Jajoda and Chandan Mazumdar (Eds.). Springer International Publishing, Cham, 98–117.

[83] A. Schaad. 2001. Detecting conflicts in a role-based delegation model. In *Seventeenth Annual Computer Security Applications Conference.* 117–126.

[84] Andreas Schaad, Volkmar Lotz, and Karsten Sohr. 2006. A Model-Checking Approach to Analysing Organisational Controls in a Loan Origination Process. In *Proceedings of the Eleventh ACM Symposium on Access Control Models and Technologies* (Lake Tahoe, California, USA) *(SACMAT '06).* Association for Computing Machinery, New York, NY, USA, 139–149. https://doi.org/10.1145/1133058.1133079

[85] A. Schaad and J. D. Moffett. 2002. A Framework for Organisational Control Principles. In *Computer Security Applications Conference, Annual.* IEEE Computer Society, Los Alamitos, CA, USA, 229. https://doi.org/10.1109/CSAC.2002.1176294

[86] Andreas Schaad and Jonathan D. Moffett. 2002. A Lightweight Approach to Specification and Analysis of Role-Based Access Control Extensions *(SACMAT '02).* Association for Computing Machinery, New York, NY, USA, 13–22. https://doi.org/10.1145/507711.507714

[87] Sigrid Schefer, Mark Strembeck, Jan Mendling, and Anne Baumgrass. 2011. Detecting and Resolving Conflicts of Mutual-Exclusion and Binding Constraints in a Business Process Context. In *On the Move to Meaningful Internet Systems: OTM 2011*, Robert Meersman, Tharam Dillon, Pilar Herrero, Akhil Kumar, Manfred Reichert, Li Qing, Beng-Chin Ooi, Ernesto Damiani, Douglas C. Schmidt, Jules White, Manfred Hauswirth, Pascal Hitzler, and Mukesh Mohania (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 329–346.

[88] M. E. Shin and G. Ahn. 2001. Role-Based Authorization Constraints Specification Using Object Constraint Language. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE Computer Society, Los Alamitos, CA, USA, 157.

[89] R.T. Simon and M.E. Zurko. 1997. Separation of duty in role-based environments. In *Proceedings 10th Computer Security Foundations Workshop*. 183–194. https://doi.org/10.1109/CSFW.1997.596811

[90] Karsten Sohr, Gail-Joon Ahn, Martin Gogolla, and Lars Migge. 2005. Specification and Validation of Authorisation Constraints Using UML and OCL. In *Computer Security – ESORICS 2005*, Sabrina de Capitani di Vimercati, Paul Syverson, and Dieter Gollmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 64–79.

[91] Karsten Sohr, Tanveer Mustafa, Xinyu Bao, and Gail-Joon Ahn. 2008. Enforcing Role-Based Access Control Policies in Web Services with UML and OCL. In *2008 Annual Computer Security Applications Conference (ACSAC)*. 257–266. https://doi.org/10.1109/ACSAC.2008.35

[92] Jon A. Solworth. 2006. Approvability *(ASIACCS '06)*. Association for Computing Machinery, New York, NY, USA, 231–242.

[93] Yunyu Song and Sylvia L. Osborn. 2006. Conflict of Interest in the Administrative Role Graph Model. In *Secure Data Management*, Willem Jonker and Milan Petković (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 100–114.

[94] Mark Strembeck and Jan Mendling. 2010. Generic Algorithms for Consistency Checking of Mutual-Exclusion and Binding Constraints in a Business Process Context. In *On the Move to Meaningful Internet Systems: OTM 2010*, Robert Meersman, Tharam Dillon, and Pilar Herrero (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 204–221.

[95] Hassan Takabi, Morteza Amini, and Rasool Jalili. 2007. Separation of Duty in Role-Based Access Control Model through Fuzzy Relations. In *Third International Symposium on Information Assurance and Security*. 125–130. https://doi.org/10.1109/IAS.2007.68

[96] Aashay Thipse and Rattikorn Hewett. 2008. Verification of Dynamic Separation of Duty Policy for Role-based Business Processes. In *2008 IEEE Region 5 Conference*. 1–6. https://doi.org/10.1109/TPSD.2008.4562752

[97] Jonathon E. Tidswell and Trent Jaeger. 2000. Integrated Constraints and Inheritance in DTAC *(RBAC '00)*. Association for Computing Machinery, New York, NY, USA, 93–102. https://doi.org/10.1145/344287.344307

[98] Jonathon E. Tidswell and John M. Potter. 1998. A dynamically typed access control model. In *Information Security and Privacy*, Colin Boyd and Ed Dawson (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 308–319.

[99] David Tranfield, David Denyer, and Palminder Smart. 2003. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. *British Journal of Management* 14 (09 2003), 207–222. https://doi.org/10.1111/1467-8551.00375

[100] J.D. Ultra and S. Pancho-Festin. 2017. A simple model of separation of duty for access control models. *Computers Security* 68 (2017), 69–80.

[101] U.S. Department of Health & Human Services. n.d.. American Health Insurance Portability and Accountability Act. https://www.hhs.gov/hipaa/index.html. Accessed: 2023-06-10.

[102] Emre Uzun, Vijayalakshmi Atluri, Haibing Lu, and Jaideep Vaidya. 2011. An Optimization Model for the Extended Role Mining Problem. In *Data and Applications Security and Privacy XXV*, Yingjiu Li (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 76–89.

[103] Simeon Veloudis and Nimal Nissanke. 2015. A Novel Permission Hierarchy for RBAC for Dealing with SoD in MAC Models. *Comput. J.* 59, 4 (08 2015), 462–492. https://doi.org/10.1093/comjnl/bxv060 arXiv:https://academic.oup.com/comjnl/article-pdf/59/4/462/7902571/bxv060.pdf

[104] Kevin W. Kobelsky. 2014. A conceptual model for segregation of duties: Integrating theory and practice for manual and IT-supported processes. *International Journal of Accounting Information Systems* 15, 4 (2014), 304–322. https://doi.org/10.1016/j.accinf.2014.05.003 2013 Research Symposium on Information Integrity Information Systems Assurance.

[105] Hua Wang, Yanchun Zhang, Jinli Cao, and Jian Yang. 2004. Specifying Role-Based Access Constraints with Object Constraint Language. In *Advanced Web Technologies and Applications*, Jeffrey Xu Yu, Xuemin Lin, Hongjun Lu, and Yanchun Zhang (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 687–696.

[106] Jane Webster and Richard Watson. 2002. Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly* 26 (06 2002).

[107] Patrick Wolf and Nick Gehrke. 2009. Continuous compliance monitoring in ERP systems-A method for identifying segregation of duties conflicts. *Wirtschaftsinformatik Proceedings 2009* 39 (2009).

[108] Christian Wolter and Andreas Schaad. 2007. Modeling of Task-Based Authorization Constraints in BPMN. In *Business Process Management*, Gustavo Alonso, Peter Dadam, and Michael Rosemann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 64–79.

[109] Christian Wolter, Andreas Schaad, and Christoph Meinel. 2007. Deriving XACML Policies from Business Process Models. In *Web Information Systems Engineering – WISE 2007 Workshops*, Mathias Weske, Mohand-Saïd Hacid, and Claude Godart (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 142–153.

[110] Benyuan Yang and Hesuan Hu. 2021. Secure Conflicts Avoidance in Multidomain Environments: A Distributed Approach. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 51, 9 (2021), 5478–5489. https://doi.org/10.1109/TSMC.2019.2954589

[111] Benyuan Yang and Hesuan Hu. 2022. Dynamic Implementation of Security Requirements in Business Processes. *IEEE Transactions on Dependable and Secure Computing* 19, 2 (2022), 1352–1363. https://doi.org/10.1109/TDSC.2020.3012729

[112] Benyuan Yang and Hesuan Hu. 2023. Analysis of Authorization Constraints via Integer Linear Programming. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2023), 2258–2271. https://doi.org/10.1109/TKDE.2021.3124271

5 A Framework for Managing Separation of Duty Policies

Current status:	Published
Conference:	19th International Conference on Availability, Reliability and Security (ARES 2024), Vienna, July 30-August 02, 2024
Date of acceptance:	02 June 2024
Full citation:	Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. A framework for managing separation of duty policies. *In Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA, 2024. Association for Computing Machine.
Authors contributions:	Sebastian Groll 50% Sascha Kern 30% Ludwig Fuchs 10% Günther Pernul 10%

Conference Description: The International Conference on Availability, Reliability and Security focuses since 2006 on rigorous and novel research in the field of dependability, computer and information security. In cooperation with the conference several workshops are held covering a huge variety of security topics.

A Framework for Managing Separation of Duty Policies

Sebastian Groll
University of Regensburg
Regensburg, Bavaria, Germany

Sascha Kern
Nexis GmbH
Regensburg, Bavaria, Germany

Ludwig Fuchs
Nexis GmbH
Regensburg, Bavaria, Germany

Günther Pernul
University of Regensburg
Regensburg, Bavaria, Germany

ABSTRACT

Separation of Duty (SoD) is a fundamental principle in information security. Especially large and highly regulated companies have to manage a huge number of SoD policies. These policies need to be maintained in an ongoing effort in order to remain accurate and compliant with regulatory requirements. In this work we develop a framework for managing SoD policies that pays particular attention to policy comprehensibility. We conducted seven semi-structured interviews with SoD practitioners from large organizations in order to understand the requirements for managing and maintaining SoD policies. Drawing from the obtained insights, we developed a framework, which includes the relevant stakeholders and tasks, as well as a policy structure that aims to simplify policy maintenance. We anchor the proposed policy structure in a generic IAM data model to ensure compatibility and flexibility with other IAM models. We then show exemplary how our approach can be enforced within Role-Based Access Control. Finally, we evaluate the proposed framework with a real-world IAM data set provided by a large finance company.

CCS CONCEPTS

• **Security and privacy → Access control**; *Usability in security and privacy*; *Formal security models*;

KEYWORDS

Separation of Duty, Identity and Access Management, Role-Based Access Control

ACM Reference Format:
Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. 2024. A Framework for Managing Separation of Duty Policies. In *The 19th International Conference on Availability, Reliability and Security (ARES 2024), July 30-August 2, 2024, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3664476.3670883

1 INTRODUCTION

The concept of Separation of Duty (SoD) is widely considered a fundamental principle in information security within organizations. SoD aims to mitigate fraud and potential conflicts of interest by dividing responsibilities and tasks among different persons. For example, a banking employee should not be able to both issue and approve a loan without at least one more person being involved in the process. Regulatory requirements such as SOX [25], HIPAA [34], BAIT [12], VAIT [6] or Basel III [4] force organisations to specify, document, and enforce SoD rules. Therefore, SoD is addressed in the academic literature concerning various contexts and research areas.

Existing research has devoted considerable attention to the specification [2, 20, 31] and enforcement [7, 9, 33] of SoD rules. However, the creation and the maintenance of SoD rules has hardly been addressed by research so far. Existing works typically assume that rules already exist or are specified by some "administrator", who knows which rules are needed. However, big organisations usually deal with thousands of employees with different functions and responsibilities leading to millions of permissions and roles in numerous departments and applications. Thus, the necessary set of SoD rules is large and complex, and the knowledge required to specify and maintain correct rules spreads throughout the organization across domain experts, managers, departmental heads, risk managers, IT specialists, etc. [18]. Once SoD rules have been initially created, they must be maintained in an ongoing effort. Similar to any set of data, SoD rules can become outdated due to changes in both external compliance requirements and the organization's internal structure (e.g. permissions or roles). Consider, for example, a new version of HIPAA, or the introduction of a new HR system in the organization. Both will lead to new requirements for the existing SoD rules. Active SoD rules must therefore be regularly reviewed, updated, and eventually de-provisioned when their validity expires.

In order to be maintainable and manageable, the meaning and purpose of each SoD rule must be clear: Where does the rule originate from? Which legal text or guideline is responsible for its existence? What happens if the legal text changes or a new one is added? Who ensures that the rule is technically implemented correctly? etc. The knowledge required to answer these questions extends throughout the entire organization. Hence, SoD rules must be created and maintained within a structured processes with clear responsibilities and knowledge holders. They need to be enriched with descriptive, human-understandable information and the knowledge spread throughout the organization must be consolidated. To the best of our knowledge, the current research does neither provide an approach for deriving and maintaining SoD rules nor does it provide an approach to enrich SoD rules with descriptive information.

ARES 2024, July 30-August 2, 2024, Vienna, Austria

ACM ISBN 979-8-4007-1718-5/24/07
https://doi.org/10.1145/3664476.3670883

This work contributes to this research gap by proposing a framework for SoD rules that are easy to maintain. Our contributions are as follows: (i) We conduct seven expert interviews with SoD managers from large, compliance-driven companies. They serve to broaden the understanding of SoD management and to identify best practices for SoD maintenance. (ii) On this basis, we propose a framework for the creation and maintenance of SoD rules, that pays particular attention to descriptive information and comprehensibility. It defines the key stakeholders for SoD maintenance, and three types of SoD policies which are anchored in the integrated IAM data model by Kunz et al. [21]. (iii) After a conceptual definition, we formalize our easy to maintain SoD rules as an exemplary proof of concept in Role-Based Access Control (RBAC) [28] and show how they can be enforced. We then evaluate the framework by applying our approach to a real-world IAM data set provided by a large finance company. Our evaluation shows that the complexity of managing and maintaining SoD with the proposed framework is significantly lower than managing traditional SoD rules.

2 FOUNDATION

SoD rules can roughly be divided into rules with static and dynamic SoD properties [14, 31]. Static rules are generally applicable, e.g. "A user who is allowed to carry out Task A must not be allowed to carry out Task B". Dynamic SoD rules are only applicable in a specific context, e.g. at the same time, with the same object or operation. An example rule would be "A user must not carry out Task A and Task B for the same customer; however, the user can carry out Task A for one customer and Task B for another". For the sake of simplicity, we'll rely on static SoD rules throughout the remainder of the work. Whenever we refer to SoD rules, we mean static SoD rules. However, we discuss how our framework can be adopted to dynamic SoD rules in section 8.

SoD is particularly important in the context of Identity and Access Management (IAM). IAM is a domain of IT management that comprises a range of processes, technologies and policies (including SoD policies), which deal with the administration of digital identities and the provisioning of secure user access to digital resources. It commonly relies on the *principle of least privilege*, which determines that no user should have more authorizations than he or she requires to perform their duties. The structure of user authorizations is defined in IAM by Access Control Models (ACMs). While various ACMs have been developed for different scenarios, some have proven to be highly adaptive in theory and practice (especially Role-Based Access Control (RBAC) [28], Attribute-Based Access Control (ABAC) [16], and Discretionary Access Control (DAC) [27]). Regardless of the utilized ACM, the defined authorizations can be expressed as a set of User Permission Assignments (UPAs) (e.g. in the form of an access matrix [29]): A permission represents an operation and a resource (e.g. "read" and "file x"). If a permission is assigned to a user, the user is authorized to perform the access specified by it. Independent of an ACM, SoD rules can be defined as sets $sod(\{p_1, ..., p_n\}, k)$ where p_i are permissions and k, n are positive integers such that $1 < k \leq n$ [22]. The rule states that there must not exist any set of users smaller than k that possess the permissions $\{p_1, ..., p_n\}$ together.

While this is a very fine-grained way to specify rules, the enforcement is proven to be computationally expensive (NP-complete) [22]. SoD is extensively studied in the context of the RBAC model, in which roles are used as intermediaries between users and permissions. A more efficient way to specify SoD rules in RBAC are Mutual Exclusive Role Policies (MER). A MER defines an amount of roles that form a toxic combination and therefore a single user is not allowed to possess together. For example the roles "Compliance Officer" and "Cashier" could form such a toxic combination, because the cashier has to carry out financial transactions and the compliance officer needs to approve them. If both roles would be assigned to the same person, this person could manipulate transactions or make false bookings. Formally, MERs are defined as sets $mer(\{r_1, ..., r_n\}, t)$ with each r_i as a role and n, t as positive integers, such that $1 < t \leq n$ [22]. In contrast to the permission-based specification, a MER policy does not allow a user to have t or more roles from the set $\{r_1, ..., r_n\}$.

MER policies are more coarse-grained than permission-based SoD policies but faster to enforce. It is also possible to translate permission-based SoD policies into MER policies. However, this will result in stricter rules that are less flexible. Other models that are addressed in SoD research include workflows (e.g. [10]), modelling languages (e.g. [30, 36]) or petri nets (e.g. [19]). Nevertheless, most approaches use an underlying RBAC model for SoD specification and enforcement. Note that the terms SoD policy and SoD rule are mostly used interchangeably, while the term constraint often describes SoD rules in the context of an ACM (e.g. "MER constraint"). We maintain this terminology in the course of this work.

3 RELATED WORK

Existing SoD literature covers a wide range of topics: The specification of SoD rules is studied in ACMs like RBAC [2, 20, 31] or ABAC [3, 5], in workflows [10, 23] or in petri nets [11, 40]. Other works enhance languages like UML [26, 32] or BPMN [37, 38] with SoD support. Another related research domain deals with the enforcement of SoD rules [7, 9, 33]. Despite broad coverage of the SoD concept in general, few works are dedicated to the actual creation of maintenance of SoD rules in the context of an organization. Some algorithmic approaches exists for the generation and transformation of SoD rules: Li et al. [8, 22] study the translation between permission-based SoD rules and MERs. As mentioned previously, the objective is to create MER policies that are minimally restrictive while still enforcing the underlying permission-based SoD policies. However, these works focus only on the algorithmic transformation of policies and not on the derivation of policies within organizations.

Kijsanayothin et al. [15] propose an approach to derive MER policies from a workflow. They assume an SoD violation when the same person is allowed to perform consecutive create and update actions on the same object. We argue that SoD rules for organizations need to be more fine-grained and flexible, for example also read-permissions might be relevant for SoD. Additionally, for some real-world applications it might be hard to define which permissions exactly grant update or create privileges on objects. Wolf and Gehrke [35] propose a 6-step method to derive SoD rules in an organization. Their method starts with an early analysis phase, e.g.

interviewing employees in order to discover and analyze relevant processes. It continues with the specification and translation of SoD rule sets and finally describes the extraction of live data from the applications and the enforcement of the specified SoD rules. While this work provides valuable insights into organizational aspects of SoD creation, its research scope is limited to Enterprise Resource Planning (ERP) systems. It provides neither a generic approach for deriving SoD rules nor semantics or a generic data structure for these rules. Furthermore, they do not map SoD rules to a standardized specification used in scientific literature.

The closely related concept of Access Control Policies (ACPs) desribes machine-processible rules which define positive or negative authorizations to regulate which access a user is allowed to make [29]. ACPs can express SoD rules, but are not limited to them. Several works in this research realm propose to align rules with human-understandable semantics to improve their maintainability. They share the assumption that the most important factor for low management complexity of rules is that they have can be associated with a real-world concept. Such a real-world concept (e.g. "all department heads may access the following resources") can be annotated to rules via attributes which represent that concept (e.g. "function = department head").

Fuchs et al. [13] highlight the high importance of annotated business semantics for effective management or maintenance of role-based ACPs. Molloy et al. [24] propose approaches to mine role-based ACPs based either on semantically meaningful user attributes or on formal concept lattices. Similarly, Jin et al. propose to assign roles to users based on user attributes [17]. Xu defines an interpretability metric which attempts to approximate the semantic meaningfulness of role-based or attribute-based ACPs by measuring their accordance with semantically meaningful user attributes [39]. To the best of our knowledge, no scientific work exists that provides a framework for the management of SoD policies in the context of an organization, or describes a generic approach to improve their human comprehensibility.

4 SEMI-STRUCTURED EXPERT INTERVIEWS

We conducted semi-structured expert interviews to gain a better understanding of how SoD rules are modeled and maintained in large organizations. We interviewed seven experts from six different organizations in accordance with the methodology proposed by Adams [1]. We prepared a questionnaire through which we went with the interviewees in a natural conversation. If further relevant points were mentioned or ambiguities arose, we deviated from the prepared catalogue in order to investigate these in more detail. The structure of the interviews can be divided into 3 blocks, which are briefly described below:

- **B1: Introduction and General Questions**
  Goal: Initiate interview. Ensure relevance of the interviewed organization.
  - Short introduction.
  - What is the participant's job position and and the participant's connection to SoD within the organization?
  - What is the organization's size? How many employees work there and how many digital identities are managed?
  - What is the main motivation for the organization to manage SoD?
- **B2: Structure and human understandability**
  Goal: Discover different SoD types and their structure. Determine how meaningful information is stored.
  - How is the organization's data model for SoD rules structured?
  - How do they make SoD understandable to people? Do they use descriptive names, or employ other attributes?
  - Which methods or tools exist in the organization to display SoDs in a human understandable way?
- **B3: Lifecycle and processes**
  Goal: Clarify stakeholders and responsible parties for SoD rule creation and maintenance. Determine how meaningful information is derived.
  - Who in the organization is responsible for SoD rule creation and maintenance?
  - How does the organization derive SoD rules?
  - How does the organization maintain SoD rules and ensure a sufficient data quality?

After conducting the interviews we structured and summarized the received answers. Naturally, there were variations in both terminology and content; for instance, some refer to roles as 'business roles' or 'organizational roles,' describing the same underlying concept. Despite these different perspectives, we could abstract several organizational structures for SoD management that are established in many or sometimes all of the considered organizations. Below we summarize significant results of the three interview blocks.

B1: The interviewed experts work on SoD projects for large organizations in different industry branches.

Organization O1 is an organization with about 17,000 employees in the engineering sector. Compared to more heavily regulated sectors, such as banking or insurance, O1 is subject to less stringent requirements. SoD rules are primarily applied to an ERP system for which a standardized set of SoD rules has been purchased. In this organization we interviewed an IAM consultant.

O2 is a large telecommunications company with about 18,000 employees. Due to both internal and external compliance requirements, SoD rules must be applied to many different application systems. They use purchased rule sets as well as self-created SoD rules. The self-created rules are derived using so-called SoD Classes, a concept we will explain further in Block B2. The interviewed expert is a risk manager.

O3 is a company with 33,000 employees. Being part of the highly regulated banking sector, it adheres to numerous legal regulations. SoD rules are created and managed using SoD Classes. Additionally, domain experts can also define pairwise mutual exclusions for roles and permissions directly. We interviewed one person working in IAM business support and one person working as a compliance manager.

O4 is an organization in the insurance sector with 7,000 employees. In the insurance sector, it is also common that compliance requirements require strong regulation through SoD. SoD rules are created and maintained solely using SoD Classes. The interviewee

Table 1: Overview of the seven interviewed SoD experts and their employing organizations.

Org.	Participants	Employees	Sector	Pairwise exclusive Permissions	Pairwise exclusive Roles	SoD Classes
O1	1	~17,000	Engineering	X		
O2	1	~18,000	Telecommunications	X		X
O3	2	~33,000	Finance	X	X	X
O4	1	~7,000	Insurance			X
O5	1	~300	Finance	X		X
O6	1	~4,500	Finance			X

emphasized that the creation of SoD rules is a complex and politically sensitive task, as it relies heavily on the support of the domain experts. We interviewed a member of the IAM team.

O5 is an organization in the financial sector with about 300 employees. It is notable that even smaller companies in the banking sector are subject to strict regulations and therefore have a high administrative burden through SoD. The organization utilizes SoD Classes and pairwise mutual exclusions for permissions, that are created and maintained by the domain experts. The SoD rules have linked legal texts and descriptions explaining why the corresponding exclusion have to exist. There also exist different risk-levels for the exclusions which are useful when SoD violations are mitigated. According to O5 not all SoD violations have to be resolved: Sometimes it is possible to grant exemptions and accept the risk. The interviewed expert is an IAM consultant.

O6 is a banking company with strict regulations like the other organizations from the finance industry. SoD rules serve to avoid conflicts of interest as well as over-privileged users, especially IT administrators. They derive SoD rules using SoD Classes. The interviewed expert is a risk management officer. Table 1 summarizes the interviewed experts and their organizations. All employee numbers were rounded to ensure the anonymity of the companies.

All of the organizations use a form of RBAC with some sort of schema definition for role hierarchies. They also manage "direct" user-permission assignments which can be granted independently from the role model, e.g. through a user self service portal. Most of the organizations operate some form of automation logic which assigns roles or permissions to users based on user attributes or proprietary rules. However, for this, none of them uses formal ABAC or the eXtensible Access Control Markup Language (XACML). The amount of managed permissions ranges between 10,000 and 1,000,000. All interviewed experts named regulatory compliance as the primary driver of their SoD management.

B2: The interviewed experts described three prototypical structures for easily understandable and maintainable SoD rules: Pairwise mutually exclusive permissions, pairwise mutually exclusive roles, and the use of SoD Classes (cf. Table 1). The simplest way is the definition of pairwise mutual exclusions between two permissions or two roles. In this case, a user is not allowed to posses both permissions or both roles at once. In contrast to the MERs defined in literature (cf. Chapter 2), which allow the definition of larger sets and a minimum number of allowed roles, the mutual exclusions in the interviewed organizations were always limited to two roles or two permissions. Such pairwise mutual exclusions are easier to understand and manage. The interviewees also stated that it is important that every mutual exclusion requires a proper and up-to-date name and description annotation to remain human-understandable. Despite their simplicity, the management of mutual exclusions has some shortcomings: They quickly amount to large data sets, causing unnecessary complexity.

Another way the experts mentioned to define SoD rules is the use of so-called SoD Classes. SoD Classes represent groupings of tasks, functions or responsibilities in an organization that are designed to conflict with each other. The SoD Classes have a name with a semantic meaning and may also have a description for further information. They are assigned to roles and permissions and a user is not allowed to posses roles or permissions with conflicting SoD Classes. According to the interviewed experts, the main benefit of SoD Classes lies in their simple management and human understandability: Typically, large organizations manage hundreds of thousands of roles and permissions, while managing only 15 to 50 SoD Classes. Another advantage is that SoD Classes are well-suited for addressing the challenge of knowledge distribution among various stakeholders regarding SoD: Legal experts can define the SoD Classes and conflicts, while technical or Domain Experts can determine the matching SoD Class for a permission or role. This becomes evident in the next section.

B3: The experts named various stakeholders which are responsible for the SoD rule creation and maintenance: Normally, the responsibility for overall SoD management lies within an IAM team. However, Domain Experts, like owners of application systems, permissions, or roles, may also be responsible, especially in large systems. For example, the IAM Team may be responsible for the entirety of SoD rules and SoD Classes, but creating mutual exclusive permissions or roles or assigning SoD Classes to permissions may be the responsibility of the permission and roles owners. Also there usually is some kind of IAM governance department, that is responsible for risk assessment and the interpretation of legal texts and compliance requirements. The creation of SoD rules typically follows one of three prototypical methods: (i) *Buying complete rule sets*, (ii) *Defining SoD classes* and (iii) *individual fine-grained rule definition*.

(i) Buying a complete rule set is a common approach to ensure compliance with regulatory requirements. Such rule sets can for example be obtained from auditing companies and cover the permissions ranged in a specific kind of target system. Since an organization does not acquire the know-how necessary to maintain these rules when purchasing, a dependency on the supplier can arise: Without regular follow-up purchases, the purchased rule sets can out-date and lose their validity over time. From a purely

compliance-driven point of view, however, they are the "safe bet": First, buying standardized rules sets that are confirmed to be compliant (at the time of purchase) gives an organization some degree of security that they will meet regulatory requirements for the covered application systems. Second, such standardized rule sets do not require to be human understandable since they are not maintained by the organization itself, but merely enforced. Therefore, those rule sets usually do not contain lots of human-understandable information. The rule sets are typically limited to one application and do not support organization-wide SoD policies. Additionally, they can also be expensive and are not available for every application, e.g. applications that are not used widely or in-house developments.

(ii) The use of SoD Classes involves multiple stakeholders: The IAM governance that defines the SoD Classes and various Domain Experts that are owners of permissions or applications. The SoD Classes are derived by the IAM governance by evaluating compliance regulations and using industry-standards. They do not usually change much over time and hence require little maintenance. The assignment of the SoD Classes to the permissions is carried out by a permission or application owners, depending on the respective application and how the application is managed. In order to keep the rules up to date, regular reviews can be carried out.

(iii) Finally, all organizations use some sort of fine-grained SoD management. This can include pairwise mutual exclusive roles and permissions, but also proprietary formats, e.g. logic-based rules or custom evaluation scripts. Managing these fine-grained SoD rules requires a deep understanding of the underlying authorization structures and has to be done by role, permission or application owners or administrators. These rules also need to be reviewed periodically to maintain their accuracy and timeliness.

5 DERIVING A FRAMEWORK FOR MANAGING AND MAINTAINING SOD

Based on the insights we gathered from the conducted expert interviews, as well as on existing SoD and IAM literature, we derived a framework for managing and maintaining SoD policies. The framework consists of three components: (i) The relevant stakeholders and their responsibilities, (ii) the three proposed SoD rule types along with the respective processes for rule creation and maintenance and (iii) the integration of the proposed SoD rule types into existing ACMs. The developed framework is presented in the remainder of this chapter. Figure 1 depicts the described tasks and responsibilities of the different stakeholders in our framework.

5.1 Stakeholders and responsibilities

In order for SoD policies to be well-understandable, they need to hold a real-world meaning. Such semantic meaningfulness does not arise from the structure of policies alone, but must be incorporated into them during their creation and maintenance. Not everyone involved in the management of SoD rules has all the necessary information available. Therefore, we first define three types of stakeholders that were described during the interviews:

- **IAM Governance / Risk Management** *(Called IAM Governance in the following)*: Responsible for complying with laws and regulatory requirements. Ensures that audit requirements are met. Manages the IAM at a high level, specifies processes and carries out risk assessments. Knows the laws and legal requirements, but does not necessarily have a deeper understanding of individual roles, permissions and IT applications.
- **IAM Team** Conducts IAM projects, is responsible for the implementation and enforcement of SoD policies and rules. Understands permissions and roles on a conceptual and technical level but cannot always assess their business implications. The IAM team size can vary from few persons up to a whole IAM Department.
- **Domain Experts** Are responsible for one or more IT application systems, processes or organizational units. Typical positions are department heads, business owners or technical owners like system admins. Domain experts know the tasks of users and the effects of permissions and roles in their responsibility in great detail. However, they do not have an overarching view of the IAM structure and may not necessarily be aware of possible interactions with other domains.

Since the necessary knowledge for creating and maintaining SoD rules is distributed across these domains, collaboration between the involved stakeholders is necessary. The IAM team serves as an intermediary between the IAM governance and the domain experts.

5.2 The SoD Matrix: A tool to reduce complexity

The SoD Matrix is a central control tool of our framework for SoD management. It acts as an intermediary between the IAM team, IAM governance and the domain experts. Preliminary for its creation is the definition of SoD classes, which are represented in the SoD matrix as rows and columns. A SoD class is an entity that describes a real-world concept relevant to SoD. An example for SoD classes could be "Payment Transactions" or "Internal Audit".

The SoD Matrix defines a pairwise exclusion whenever two SoD classes are incompatible with one another. Due to its Matrix form, the SoD Matrix is intuitively understandable and suitable for visualization. Figure 2 displays a (reduced) SoD Matrix as managed by a real-world finance company. The definition and maintenance of the SoD Matrix (and SoD classes) is carried out by the IAM governance through reading and interpreting legal texts and security standards. The IAM Governance has the required expertise and knowledge for this task. A deeper understanding of the applications or their permissions and roles is not required. For example, a member of IAM Governance could read in a compliance standard that functions related to payment transactions must not be combined with functions dealing with compliance and approval. The IAM governance employee would create the SoD classes "Payment Transactions" and "Internal Audit" as well as an exclusions between the two SoD classes in the SoD matrix.

In order to enforce the SoD matrix on the application level, the SoD classes need to be assigned to the permissions of the respective applications. This task is carried out by the domain experts, as it requires in-depth knowledge of the permissions, particularly regarding which actions can be performed with which permissions. At this point, it is crucial that the SoD classes have meaningful

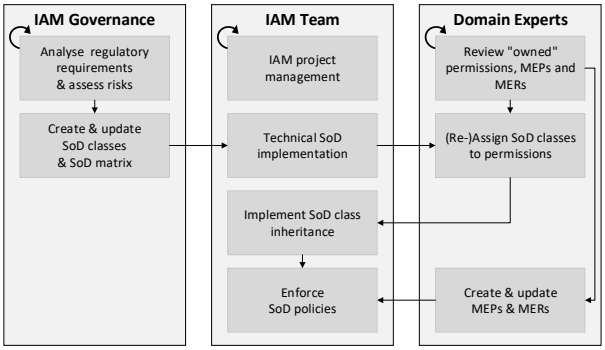


Figure 1: Responsibilities and activities of the proposed framework.

names and descriptions to be understood by the domain experts. For example a permission that allows a user to edit a financial transaction on an ERP system would be assigned to a SoD class "Payment Transactions" by a domain expert. Note that many permissions are not relevant for SoD (e.g. "WiFi access"). All permissions without a matching SoD class are assigned a neutral SoD class which can never cause a SoD conflict with another SoD class.

Applying the SoD Matrix to all permissions results in a list of pairwise mutual permission exclusions. The IAM team must bring the SoD Matrix and the permissions with SoD classes together and enforce the resulting exclusions. These can be implemented according to any established SoD specification (cmp. section 2). Maintaining the resulting (technical) SoD rules no longer requires involving governance or domain experts. In return, governance and domain experts can keep the SoD components in their responsibility up-to-date on their own, without having to rely on the knowledge of others. In addition to the IAM project organization required for the creation of a SoD Matrix, the IAM team must provide the technical infrastructure and organize regular reviews. The described responsibilities and tasks can be seen in Figure 1.

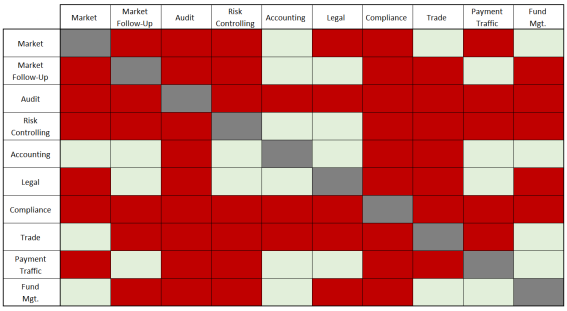


Figure 2: Matrix visualization of SoD classes and their pairwise mutual exclusions ("SoD Matrix").
Note: Red cells mark a pairwise exclusion, while green cells indicate that the two SoD classes are not conflicting. The depicted SoD Matrix defines 31 pairwise SoD class exclusions.

5.3 Fine-grained exclusions with pairwise MERs and MEPs

The SoD matrix enables well-maintainable and compliant, but coarse-grained SoD rules. In real-world applications, more precise SoD rules may be necessary. For example, two specific permissions (or roles) may form a toxic combination, but be included in compatible SoD classes. This can happen for technical reasons, especially since permissions function slightly differently in each application. Also there may not be a suitable SoD class for some roles.

The knowledge about these exclusions lies with the domain experts. With pairwise Mutually Exclusive Roles (MER) and pairwise Mutually Esclusive Permissions (MEP) they can create direct exclusions, in addition to the SoD matrix. In contrast to MERs specified in scientific literature, we propose using MERs and MEPs that require a mandatory description for each exclusion explaining its respective reason. Furthermore, they have a fixed cardinally of 2, which makes pairwise MERs an instance of the default RBAC MER: $mer_2\{r1, r2\} \equiv mer\langle\{r_1, r_2\}, 2\rangle$. The processes for creating and maintaining MERs and MEPs are performed by the domain experts and organized by the IAM team. Again the IAM team is responsible for providing a suitable infrastructure and organizing regular reviews.

5.4 Data model integration

Existing IAM literature defines numerous ACMs designed for different scenarios and requirements. Among the most commonly used ACMs are RBAC, ABAC and the somewhat older but still relevant DAC model. Alongside these theoretical data structures exist industry standards for identity data exchange, SSO and authorization delegation. Since a precise specification of the proposed SoD policies in a significant amount of ACMs and standards exceeds the scope of this work, we chose to anchor it in the conceptual IAM data model proposed by Kunz et al [21]. It provides a generic view over the central entities defined in the ACMs, e.g. RBAC and ABAC, and in the standards LDAP, SAML, SPML, OAuth, SCIM and XACML. To the best of our knowledge, it is the most generic IAM data model proposed in scientific literature. A more detailed

formalization of the proposed SoD policies for the RBAC model is provided in section 6.

The conceptual data model defines a digital identity as a representation of a human user. An account is a representation of a user within an application system and can hence be part of a digital identity. Permissions enable users to execute certain actions, commonly expressed as a combination of a resource to be accessed and an access operation (e.g. modify files in a certain directory). Roles are semantic entities that bundle a set of permissions. They are valid organization-wide and can hence be assigned directly to a digital identity. Policies represent sets of rules that define authorizations, such as ABAC, SPML or XACML. Attributes represent meta-data that is assigned to any of the described entities. An example of a user attribute would be their job title or the department they work in. The original data model also defines a context entity, which we omit because it is not relevant for the scope of this work.

We extend the conceptual model by Kunz et al.[21] to cover the proposed SoD policies. SoD policy is a specific type of policy that constrains the authorization structure by defining mutually exclusive entitlements. A violation of this constraint represents a conflict that must be resolved. We define three kinds of SoD polices: Pairwise MERs, MEPs and the SoD matrix (along with its SoD classes). Pairwise MER and MEP policies, as specified before, define simple pairs of mutually exclusive roles or permissions. They are hence linked to exactly two permissions (MEP) or roles (MER) each and restrict their assignment to a digital identity: Any pair of mutually exclusive permissions or roles that is inherited by a single digital identity constitutes a violation of the defining policy which must be resolved to restore compliance.

The SoD matrix defines mutually exclusive SoD classes. The SoD classes are assigned to permissions as an attribute. Every permission has exactly one SoD class assigned, which may also be the neutral SoD class that indicates no SoD relevance. Roles inherit SoD classes transitively from their permissions. We constrain the model by demanding that roles are *homogeneous*, i.e. must not inherit permissions with more than one SoD class other than the neutral one. Each pair of mutually exclusive SoD classes in the SoD matrix indicates that no digital identity may inherit permissions from both of these classes. Figure 3 summarizes the SoD policies from the proposed framework, anchored in the conceptual IAM model.

6 POLICY FORMALIZATION AND ENFORCEABILITY IN RBAC

We extended the conceptual IAM model by Kunz et al. to anchor the SoD policies in a generic data model. In the course of this chapter, we will provide a more detailed formalization in the RBAC model to demonstrate its feasibility. We then show that the proposed SoD policies can be translated into classic MER constraints. We chose RBAC because it is a well-known ACM in SoD literature as well as in the organizations of the interviewed SoD experts. A practical evaluation is provided subsequently in section 7.

6.1 Model formalization

We adopt the preliminaries and notations from the RBAC SoD formalizations by Li et al. [22] for sakes of consistency.

Preliminaries. Let U be the set of all users, R the set of all roles and P the set of all permissions. An RBAC state γ is a 3-tuple $\langle UA, PA, RH\rangle$, where $UA \subset U \times R$ is the set of all user-to-role assignments, $PA \subset R \times P$ is the set of all role-to-permission assignments, and $RH \subset R \times R$ is the set of all role-to-role assignments. RH^* is the reflexive, transitive closure of RH and a partial order among roles in R. The functions $auth\_roles_\gamma[u]$ and $auth\_perms_\gamma[u]$ determine the roles and permissions that a user effectively inherits and are defined as:

$$auth\_roles_\gamma[u] = \{r \in R | \exists r_1 \in R[(u, r_1) \in UA \wedge (r_1, r) \in RH^*]\}$$

$$auth\_perms_\gamma[u] = \{p \in P | \exists r_1, r_2 \in R[(u, r_1) \in UA \wedge (r_1, r_2) \in RH^* \wedge (r_2, p) \in PA]\}$$

Definition 1. A pairwise MER policy mer_2 is a specific type of MER constraint with fixed value of $n = t = 2$. It defines a pair of mutually exclusive roles

$$mer_2\{r_1, r_2\}$$

with $r_1, r_2 \in R$ and $r_1 \neq r_2$. A pairwise MER policy is satisfied by an RBAC state γ if no single user inherits both roles defined by it as mutually exclusive:

$$sat_{mer_2}[\gamma] \triangleq \forall u \in U(|auth\_roles_\gamma[u] \cap \{r_1, r_2\}| < 2)$$

Definition 2. A pairwise MEP policy mep_2 defines a pair of mutually exclusive permissions

$$mep_2\{p_1, p_2\}$$

with $p_1, p_2 \in P$ and $p_1 \neq p_2$. It is satisfied by an RBAC state γ if no single user inherits both permissions defined by it as mutually exclusive:

$$sat_{mep_2}[\gamma] \triangleq \forall u \in U(|auth\_perms_\gamma[u] \cap \{p_1, p_2\}| < 2)$$

Definition 3. An organization manages a finite set of $I + 1$ SoD classes with $\emptyset$ being the *neutral SoD class*:

$$SOD\_CLASSES = \{\emptyset, class_1, ..., class_I\}$$

Every permission p is assigned a SoD class $assigned\_class_p[p] \in SOD\_CLASSES$. By default, any permission has $assigned\_class_p[p] = \emptyset$, unless it was assigned a non-neutral $class \in \{class_1, ..., class_I\}$. The set of all permissions which share a SoD class $class$ is determined by the function

$$perms[class] = \{p \in P | assigned\_class_p[p] = class\}$$

A role inherits SoD classes transitively from its child roles and permissions:

$$trans\_class[r] = \{class \in SOD\_CLASSES | \exists r_1 \in R[(r, r_1) \in RH^* \wedge (r_1, p) \in PA \wedge assigned\_class_p[p] = class]\}$$

The effectively assigned SoD classes are consequently:

$$assi\_class_r[r] = \begin{cases} \{\emptyset\} & \text{if } trans\_class[r] = \{\emptyset\} \\ & \text{or } trans\_class[r] = \{\} \\ trans\_class[r] \setminus \{\emptyset\} & \text{otherwise} \end{cases}$$

We call an RBAC state γ *homogeneous* if no role inherits more than one non-neutral SoD class:

$$homogeneous[\gamma] \triangleq |assi\_class_r[r]| = 1 \forall r \in R$$

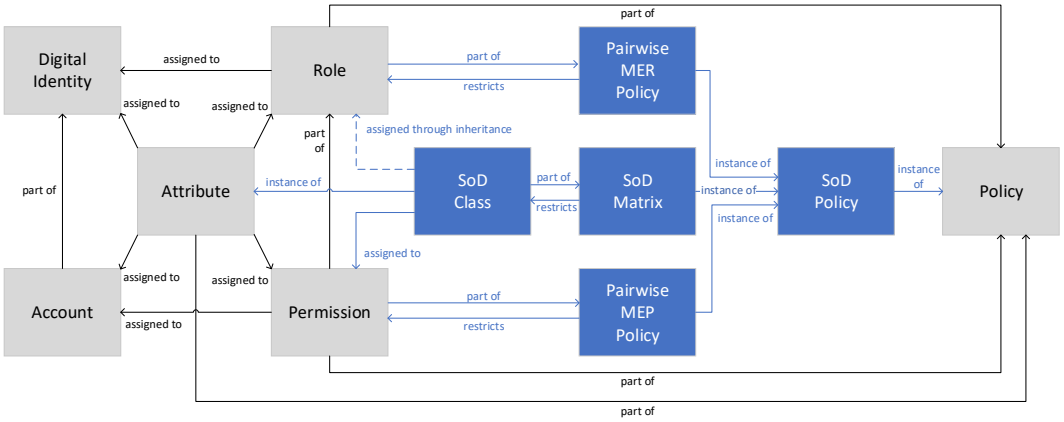


Figure 3: Anchoring of proposed SoD policies in the conceptual IAM data model by Kunz et al. [21].
Note: Grey entities were adopted from the original conceptual IAM model, while blue entities are introduced in the proposed SoD framework.

The set of all roles which share a SoD class *class* is determined by the function

$$roles[class] = \{r \in R | class \in assi\_class_r[r]\}$$

Definition 4. A user inherits all SoD classes of her assigned roles, except the neutral one:

$$assigned\_class_u[u] = \{class \in SOD\_CLASSES \setminus \{\varnothing\} |$$
$$\exists r_1 \in R[r_1 \in auth\_roles_\gamma[u] \wedge assi\_class_r[r_1] = class]\}$$

Definition 5. A SoD matrix is a set of pairwise mutual SoD class exclusions. A pairwise mutual SoD class exclusion mec_2 defines a pair of mutually exclusive SoD classes

$$mec_2\{class_1, class_2\}$$

with $class_1, class_2 \in SOD\_CLASSES \setminus \varnothing$ and $class_1 \neq class_2$. It is satisfied by a homogeneous RBAC state γ if no single user inherits roles which are assigned both SoD classes defined by it as mutually exclusive:

$$sat_{mec_2}[\gamma] \triangleq \forall u \in U(|assigned\_class_u[u]| < 2)$$

Note that we intentionally limit the use of SoD policies to homogeneous RBAC states to improve role semantics. While the model formalization would be equally applicable for non-homogeneous RBAC states, we argue that an RBAC state can be made homogeneous relatively easy, which improves the overall policy understandability and maintainability (cmp. sections 5 and 8).

6.2 Policy Enforcement

In the following we show that the proposed SoD policies are compatible with established SoD specifications for RBAC. In RBAC, SoD rules are commonly specified as MER constraints $mer\langle\{r_1, ..., r_n\}, t\rangle$, with each r_i as a role and n, t as positive integers, such that $1 < t \leq n$. If t or more roles are assigned to a user, the rule is violated. Formally, an RBAC state satisfying an MER constraint is denoted as follows [22]:

$$sat_{mer}[\gamma] \triangleq \forall u \in U(|auth\_roles_\gamma[u] \cap \{r_1, ...r_n\}| < t)$$

Thus, to show how the proposed SoD policies can be enforced in RBAC, we have to show how the three types of SoD policies (pairwise MEP, MER and mutually exclusive SoD classes) can be translated to MER constraints, as defined above. This is trivial for the pairwise MER policy since it can be defined as MER with $n = t = 2$:

$$mer_2\{r_1, r_2\} \equiv mer\langle\{r_1, r_2\}, 2\rangle$$

A pairwise MEP policy is equivalent to the set of pairwise MER policies that defines pairwise exclusions for all roles which inherit the mutually exclusive permissions transitively:

$$mep_2\{p_1, p_2\} \equiv \{mer_2\{r_1, r_2\} |$$
$$\forall r_1 \in R[\exists r \in R | (r_1, r) \in RH^* \wedge (r, p_1) \in PA],$$
$$r_2 \in R[\exists r \in R | (r_2, r) \in RH^* \wedge (r, p_2) \in PA]\}$$

A pairwise mutual SoD class exclusion is equivalent to the set of pairwise MER policies that defines pairwise exclusions for all roles with the specified SoD classes:

$$mec_2\{class_1, class_2\} \equiv \{mer_2\{r_1, r_2\} |$$
$$\forall r_1 \in \text{roles}[class_1], r_2 \in \text{roles}[class_2]\}$$

Both pairwise MEPs and mutual SoD class exclusions can be translated into a set of pairwise MER policies. Since any pairwise MER policy can be expressed as a classic MER constraint, any SoD matrix can be translated into established MER constraints and enforced as such.

7 EVALUATION

We cooperated with a large financial services provider to evaluate the proposed framework. An IAM expert from the company also took part in the expert interviews (see organization O6 in table 1). The conceptual overview of stakeholders and responsibilities was generally agreed upon as it was synthesized from the expert interviews. We hence defined two evaluation criteria: (i) *Technical applicability*: Can we confirm that the proposed SoD matrix and mutual pairwise exclusions are compatible with existing SoD

specifications and can be enforced as such? (ii) *Simplified policy management*: Can we confirm that the framework provides a complexity reduction compared to established MER policies?

We were allowed to analyze anonymized, productively used SoD entitlement data of the organization. O6 employs around 4,500 people and manages around 10,000 digital identities. Similar to the other companies in the expert interviews, the banking group organizes SoD policies through SoD classes and a SoD matrix. The permissions in the data set are organized in the form of an RBAC model with a flat role hierarchy: The roles are called "business roles". Business roles are valid in the context of the entire company and bundle users based on a functional assignment, e.g. “customer support” or “liquidity forecast”. The permissions are organized in two layers: "system entitlements" and "technical permissions". Technical permissions are fine-grained permissions that are only assigned to system entitlements. SoD classes are only assigned to system entitlements, which makes the technical permissions irrelevant for our use case. Hierarchy relations between two business roles or two system entitlements are not allowed. We analyzed a total of 14 SoD classes and about 8,000 system entitlements.

We implemented two algorithms that allow us to generate MERs based on the SoD matrix, SoD classes and system entitlements. While we could not include them in this work due to space constraints, we provide them on a GitHub repository along with sample data to execute them[1]. The first algorithm collects SoD classes from permissions and transitively writes them to the inheriting roles. It guarantees the roles’ SoD class homogeneity and marks roles that would otherwise inherit more than one SoD class. If a role would inherit more than one SoD class we assign ×, which represents an invalid SoD class state. These violations can be addressed by splitting the permissions of the role into multiple roles. The second algorithm transforms a SoD Matrix and a set of homogeneous roles into equivalent MER constraints.

By applying the algorithms on the data, we assigned at least one SoD class to 209 roles. Out of these, 5 had more than one SoD class. Based on these assignments, we generated a total of 12,295 MERs. Table 2 summarizes the results. We validated the results with the IAM experts from O6. The experts confirmed the correctness of our results and the usefulness of our approach. They pointed out that the effort and error rate in managing SoD classes is significantly lower compared to managing MERs. MERs, however, can be enforced more easily and are immediately compatible with common IAM systems. On this basis we conclude that the evaluation criteria were met:

(i) *Technical applicability*: We were able transform a SoD matrix and permissions assigned with SoD classes of a real-world banking group into classic MERs. This means that a SoD matrix can be enforced as such. The enforcement of pairwise MERs and MEPs is trivial and the correctness of our translation was confirmed by the IAM experts of O6.

(ii) *Simplified policy management*: The quantifiable data complexity required to manage a SoD matrix is significantly lower than the data complexity for equivalent MERs. The real-world data set contained 14 SoD classes, 32 pairwise SoD class exclusions, 274 permissions with a non-neutral SoD class and 209 roles with at least one non-neutral SoD-class, amounting to a total of 529 entities that need to be managed. In contrast, an equivalent set of pairwise MERs amounts to 12,295 entities. In theory, the MERs could be algorithmically optimized to cover the same exclusions with fewer MERs that have a higher cardinality. However, these MERs would be optimized for low data volume only and would not represent a human-understandable real-world concept. Note also that individual stakeholders have to manage fewer than 529 entities: Governance employees only need to maintain the SoD matrix with 14 SoD classes and 32 pairwise exclusions, and domain experts only manage permissions, of which only 274 have a non-neutral SoD class.

[1]https://github.com/sgroll/semantic_sod

Table 2: Output of algorithm execution on real world data

Basic data	
Roles	2,494
Permissions	7,972
Role-permission assignments	18,692
SoD class data	
SoD classes	14
Pairwise SoD class exclusions (SoD Matrix)	32
Permissions with non-neutral SoD class	274
Roles with non-neutral SoD class	209
Algorithm results	
Resulting MERs	12,295
Homogeneity violations	5

8 DISCUSSION & CONCLUSION

This work proposed a framework for the creation and maintenance of semantically meaningful SoD policies. The work is grounded on seven expert interviews with SoD practitioners from the industry. In these interviews we attempted to gain an understanding and summarize best practices for SoD management in large and compliance-driven organizations. The developed framework aims to provide structures for semantically meaningful SoD which remain aligned with established practices.

The framework starts with a definition of SoD stakeholders and their responsibilities. It then defines thee types of SoD policies which are designed to be well-maintainable and easily understandable. Finally, the defined SoD policies are anchored in an integrated IAM data model to generalize their scope. After presenting the proposed framework, this work provides a formalization of the three defined SoD policy types in the well-established RBAC model. We show that the policies can be translated into classic MER constraints and evaluate this with a real-world data set, thus proving that they can be evaluated and enforced efficiently.

The proposed SoD policies are easily maintainable and understandable because of three characteristics: (i) All managed policies have a semantic meaning derived from a real-world requirement. (ii) The SoD matrix decouples the creation of SoD exclusions from the management of permissions, which eliminates the need for collaboration between stakeholders from different knowledge domains. (iii) The quantifiable data complexity of SoD exclusions managed

via the SoD matrix is significantly lower than the complexity of equivalent MERs. Our work currently only applies to rules with static SoD properties. This can be addressed by developing dynamic SoD classes: Similar to dynamic SoD rules, a user may posses permissions of conflicting SoD classes. When permissions of one of the two classes are used within a specified context (e.g. time, operation, customer etc.), the permissions of the other class cannot be used in this context. We leave a detailed specification and analyses of dynamic SoD classes for future work.

ACKNOWLEDGMENTS

The research leading to these results was supported by the German Federal Ministry of Education and Research as part of the DEVISE project (https://devise.ur.de/).

REFERENCES

[1] William C Adams. 2015. Conducting semi-structured interviews. *Handbook of practical program evaluation* (2015), 492–505.
[2] Gail-Joon Ahn and Ravi Sandhu. 2000. Role-Based Authorization Constraints Specification. 3, 4 (nov 2000), 207–226. https://doi.org/10.1145/382912.382913
[3] Nuray Baltaci and James Joshi. 2020. A Constraint and Risk-aware Approach to Attribute-based Access Control for Cyber-Physical Systems. *Computers & Security* 96 (05 2020), 101802. https://doi.org/10.1016/j.cose.2020.101802
[4] Basel Committee on Banking Supervision. 2011. *Basel III - A Global Regulatory Framework for More Resilient Banks and Banking Systems.* Bank for International Settlements.
[5] Khalid Bijon, R. Krishman, and R. Sandhu. 2013. Constraints specication in attribute based access control. *SCIENCE* 2 (01 2013), 131–144.
[6] Bundesanstalt für Finanzdienstleistungsaufsicht. 2024. *Versicherungsaufsichtliche Anforderungen an die IT (VAIT).*
[7] David W Chadwick, Wensheng Xu, Sassa Otenko, Romain Laborde, and Bassem Nasser. 2007. Multi-session Separation of Duties (MSoD) for RBAC. In *2007 IEEE 23rd International Conference on Data Engineering Workshop.* 744–753. https://doi.org/10.1109/ICDEW.2007.4401062
[8] Hong Chen and Ninghui Li. 2006. Constraint Generation for Separation of Duty *(SACMAT '06).* Association for Computing Machinery, New York, NY, USA, 130–138. https://doi.org/10.1145/1133058.1133077
[9] Weihe Chen, Zhu Tang, and Shiguang Ju. 2008. Enforcement of Spatial Separation of Duty Constraint. In *2008 The 9th International Conference for Young Computer Scientists.* 2108–2114. https://doi.org/10.1109/ICYCS.2008.223
[10] Jason Crampton. 2005. A Reference Monitor for Workflow Systems with Constrained Task Execution *(SACMAT '05).* Association for Computing Machinery, New York, NY, USA, 38–47. https://doi.org/10.1145/1063979.1063986
[11] Marcelo Antonio de Carvalho and Paulo Bandiera-Paiva. 2017. Evaluating ISO 14441 privacy requirements on role based access control (RBAC) restrict mode via Colored Petri Nets (CPN) modeling. In *2017 International Carnahan Conference on Security Technology (ICCST).* 1–8. https://doi.org/10.1109/CCST.2017.8167833
[12] Deutsche Bundesbank. 2017. Bankaufsichtliche Anforderungen an die IT (BAIT). Available at https://www.bundesbank.de/de/aufgaben/bankenaufsicht/einzelaspekte/risikomanagement/bait/bankaufsichtliche-anforderungen-an-die-it-598580.
[13] Ludwig Fuchs, Michael Kunz, and Günther Pernul. 2014. Role model optimization for secure role-based identity management. (2014).
[14] Virgil D Gligor, Serban I Gavrila, and David Ferraiolo. 1998. On the formal definition of separation-of-duty policies and their composition. In *Proceedings. 1998 IEEE Symposium on Security and Privacy (Cat. No. 98CB36186).* IEEE, 172–183.
[15] Rattikorn Hewett, Phongphun Kijsanayothin, and Aashay Thipse. 2008. Security analysis of role-based separation of duty with workflows. In *2008 Third International Conference on Availability, Reliability and Security.* IEEE, 765–770.
[16] Vincent Hu, David Ferraiolo, D. Kuhn, A. Schnitzer, Knox Sandlin, R. Miller, and Karen Scarfone. 2014. Guide to attribute based access control (ABAC) definition and considerations. *National Institute of Standards and Technology Special Publication* (01 2014), 162–800.
[17] Xin Jin, Ram Krishnan, and Ravi Sandhu. 2012. A role-based administration model for attributes. In *Proceedings of the first international workshop on secure and resilient architectures and systems.* 7–12.
[18] Sascha Kern, Thomas Baumer, Ludwig Fuchs, and Günther Pernul. 2023. Maintain High-Quality Access Control Policies: An Academic and Practice-Driven Approach. In *IFIP Annual Conference on Data and Applications Security and Privacy.* Springer, 223–242.
[19] Konstantin Knorr and Harald Weidner. 2001. Analyzing Separation of Duties in Petri Net Workflows. In *Information Assurance in Computer Networks*, Vladimir I. Gorodetski, Victor A. Skormin, and Leonard J. Popyack (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 102–114.
[20] D. Richard Kuhn. 1997. Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems. In *Proceedings of the Second ACM Workshop on Role-Based Access Control* (Fairfax, Virginia, USA) *(RBAC '97).* Association for Computing Machinery, New York, NY, USA, 23–30. https://doi.org/10.1145/266741.266749
[21] Michael Kunz, Alexander Puchta, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. 2019. Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications* 44 (2019), 64–79. https://doi.org/10.1016/j.jisa.2018.11.004
[22] Ninghui Li, Mahesh V. Tripunitara, and Ziad Bizri. 2007. On mutually exclusive roles and separation-of-duty. *ACM Trans. Inf. Syst. Secur.* 10, 2 (may 2007), 5–es. https://doi.org/10.1145/1237500.1237501
[23] Jan Mendling, Karsten Ploesser, and Mark Strembeck. 2008. Specifying Separation of Duty Constraints in BPEL4People Processes. In *Business Information Systems*, Witold Abramowicz and Dieter Fensel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 273–284.
[24] Ian Molloy, Hong Chen, Tiancheng Li, Qihua Wang, Ninghui Li, Elisa Bertino, Seraphin Calo, and Jorge Lobo. 2008. Mining roles with semantic meanings. In *Proceedings of the 13th ACM symposium on Access control models and technologies.* 21–30.
[25] One Hundred Seventh Congress of the United States of America. 2002. Sarbanes-Oxley Act of 2002. Pub. L. No. 107-204, 116 Stat. 745.
[26] Indrakshi Ray, Na Li, Robert France, and Dae-Kyoo Kim. 2004. Using Uml to Visualize Role-Based Access Control Constraints *(SACMAT '04).* Association for Computing Machinery, New York, NY, USA, 115–124. https://doi.org/10.1145/990036.990054
[27] Pierangela Samarati and Sabrina Capitani de Vimercati. 2000. Access control: Policies, models, and mechanisms. In *International school on foundations of security analysis and design.* Springer, 137–196.
[28] Ravi S Sandhu. 1998. Role-based access control. In *Advances in computers.* Vol. 46. Elsevier, 237–286.
[29] Ravi S Sandhu and Pierangela Samarati. 1994. Access control: principle and practice. *IEEE communications magazine* 32, 9 (1994), 40–48.
[30] M. E. Shin and G. Ahn. 2001. Role-Based Authorization Constraints Specification Using Object Constraint Language. In *2012 IEEE 21st International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises.* IEEE Computer Society, Los Alamitos, CA, USA, 157. https://doi.org/10.1109/ENABL.2001.953406
[31] R.T. Simon and M.E. Zurko. 1997. Separation of duty in role-based environments. In *Proceedings 10th Computer Security Foundations Workshop.* 183–194. https://doi.org/10.1109/CSFW.1997.596811
[32] Karsten Sohr, Gail-Joon Ahn, Martin Gogolla, and Lars Migge. 2005. Specification and Validation of Authorisation Constraints Using UML and OCL. In *Computer Security – ESORICS 2005*, Sabrina de Capitani di Vimercati, Paul Syverson, and Dieter Gollmann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 64–79.
[33] Karsten Sohr, Tanveer Mustafa, Xinyu Bao, and Gail-Joon Ahn. 2008. Enforcing Role-Based Access Control Policies in Web Services with UML and OCL. In *2008 Annual Computer Security Applications Conference (ACSAC).* 257–266. https://doi.org/10.1109/ACSAC.2008.35
[34] U.S. Department of Health & Human Services. n.d.. American Health Insurance Portability and Accountability Act. https://www.hhs.gov/hipaa/index.html. Accessed: 2023-06-10.
[35] Patrick Wolf and Nick Gehrke. 2009. Continuous compliance monitoring in ERP systems-A method for identifying segregation of duties conflicts. *Wirtschaftsinformatik Proceedings 2009* 39 (2009).
[36] Christian Wolter and Andreas Schaad. 2007. Modeling of Task-Based Authorization Constraints in BPMN. In *Business Process Management*, Gustavo Alonso, Peter Dadam, and Michael Rosemann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 64–79.
[37] Christian Wolter and Andreas Schaad. 2007. Modeling of Task-Based Authorization Constraints in BPMN. In *Business Process Management*, Gustavo Alonso, Peter Dadam, and Michael Rosemann (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 64–79.
[38] Christian Wolter, Andreas Schaad, and Christoph Meinel. 2007. Deriving XACML Policies from Business Process Models. In *Web Information Systems Engineering – WISE 2007 Workshops*, Mathias Weske, Mohand-Saïd Hacid, and Claude Godart (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 142–153.
[39] Zhongyuan Xu. 2014. *Mining meaningful role-based and attribute-based access control policies.* Ph. D. Dissertation. State University of New York at Stony Brook.
[40] Benyuan Yang and Hesuan Hu. 2023. Analysis of Authorization Constraints via Integer Linear Programming. *IEEE Transactions on Knowledge and Data Engineering* 35, 3 (2023), 2258–2271. https://doi.org/10.1109/TKDE.2021.3124271

Bibliography

[1] Basel Committee on Banking Supervision. *Basel III - A Global Regulatory Framework for More Resilient Banks and Banking Systems*. Bank for International Settlements, 2011.

[2] Victor R Basili. Applying the goal/question/metric paradigm in the experience factory. *Software quality assurance and measurement: A worldwide perspective*, 7(4):21–44, 1993.

[3] Rakesh Bobba, Serban Gavrila, Virgil Gligor, Himanshu Khurana, and Radostina Koleva. Administering access control in dynamic coalitions. In *Proceedings of the 19th Conference on Large Installation System Administration Conference - Volume 19*, LISA '05, page 23, USA, 2005. USENIX Association.

[4] Bundesanstalt für Finanzdienstleistungsaufsicht. Versicherungsaufsichtliche anforderungen an die it (VAIT). Available at https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Rundschreiben/2018/rs_18_10_vait_va.html, 2024.

[5] David D. Clark and David R. Wilson. A comparison of commercial and military computer security policies. In *1987 IEEE Symposium on Security and Privacy*, pages 184–184, 1987.

[6] Stefan Cronholm and Hannes Göbel. Action design research: integration of method support. *International Journal of Managing Projects in Business*, 15(8):19–47, 2022.

[7] Deutsche Bundesbank. Bankaufsichtliche anforderungen an die it (BAIT). Available at https://www.bundesbank.de/de/aufgaben/bankenaufsicht/einzelaspekte/risikomanagement/bait/bankaufsichtliche-anforderungen-an-die-it-598580, 2017.

[8] David Ferraiolo and David Kuhn. Role-based access controls. 15th National Computer Security Conference, Baltimore, MD, 1992-10-13 1992.

[9] Tomer Geva and Maytal Saar-Tsechansky. Who's a good decision maker? data-driven expert worker ranking under unobservable quality. In *2016 International Conference on Information Systems, ICIS 2016*, 2016 International Conference on Information Systems, ICIS 2016. Association for Information Systems, 2016.

[10] Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Separation of duty in information security. *ACM Computing Surveys*, N/A.

[11] Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. Monitoring access reviews by crowd labelling. In Simone Fischer-Hübner, Costas Lambrinoudakis, Gabriele Kotsis, A. Min Tjoa, and Ismail Khalil, editors, *Trust, Privacy and Security in Digital Business*, pages 3–17, Cham, 2021. Springer International Publishing.

[12] Sebastian Groll, Sascha Kern, Ludwig Fuchs, and Günther Pernul. A framework for managing separation of duty policies. In *Proceedings of the 19th International Conference on Availability, Reliability and Security*, ARES '24, New York, NY, USA, 2024. Association for Computing Machinery.

[13] Alan Hevner and Samir Chatterjee. *Design Research in Information Systems – Theory and Practice*, chapter Design Science Research in Information Systems, pages 9–22. Springer US, Boston, MA, 2010.

[14] Alan Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design Science in Information Systems Research. *Management Information Systems Quarterly*, 28(1):75–105, 2004.

[15] Vincent C. Hu, David F. Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller, and Karen Scarfone. Guide to attribute based access control (abac) definition and considerations. Technical Report 800-162, National Institute of Standards and Technology (NIST), 2014.

[16] Pooya Jaferian, Hootan Rashtian, and Konstantin Beznosov. To authorize or not authorize: helping users review access policies in organizations. In *10th Symposium On Usable Privacy and Security ({SOUPS} 2014)*, pages 301–320, 2014.

[17] Sascha Kern, Thomas Baumer, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Optimization of access control policies. *Journal of Information Security and Applications*, 70:103301, 2022.

[18] Michael Kunz, Alexander Puchta, Sebastian Groll, Ludwig Fuchs, and Günther Pernul. Attribute quality management for dynamic identity and access management. *Journal of Information Security and Applications*, 44:64–79, 2019.

[19] Ninghui Li, Mahesh V. Tripunitara, and Ziad Bizri. On mutually exclusive roles and separation-of-duty. *ACM Trans. Inf. Syst. Secur.*, 10(2):5–es, may 2007.

[20] One Hundred Seventh Congress of the United States of America. Sarbanes-oxley act of 2002, 2002. Pub. L. No. 107-204, 116 Stat. 745.

[21] Federal office for information security (BSI). It-grundschutz. https://www.bsi.bund.de/DE/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/IT-Grundschutz/it-grundschutz_node.html, n.d. Accessed: 2023-06-10.

[22] Chitu Okoli and Kira Schabram. A guide to conducting a systematic literature review of information systems research. *SSRN Electronic Journal*, 10, 05 2010.

[23] Andreas Pfitzmann and Marit Hansen. A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management. 2010.

[24] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. Role-Based Access Control Models. *Computer*, 29(2):38–47, 1996.

[25] Maung K Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. Action design research. *MIS quarterly*, pages 37–56, 2011.

[26] U.S. Department of Health & Human Services. American health insurance portability and accountability act. https://www.hhs.gov/hipaa/index.html, n.d. Accessed: 2023-06-10.

[27] Yair Wand and Richard Y. Wang. Anchoring data quality dimensions in ontological foundations. *Commun. ACM*, 39(11):86–95, nov 1996.

[28] Jane Webster and Richard Watson. Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly*, 26, 06 2002.

[29] Robert Winter. Design science research in europe. *European Journal of Information Systems*, 17:470–474, 10 2008.